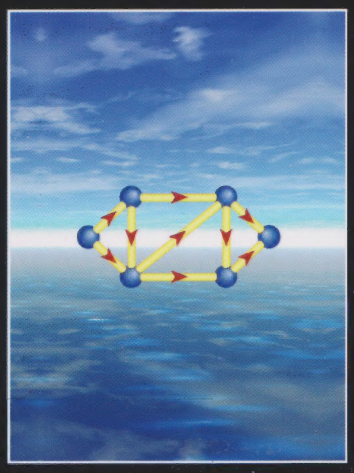
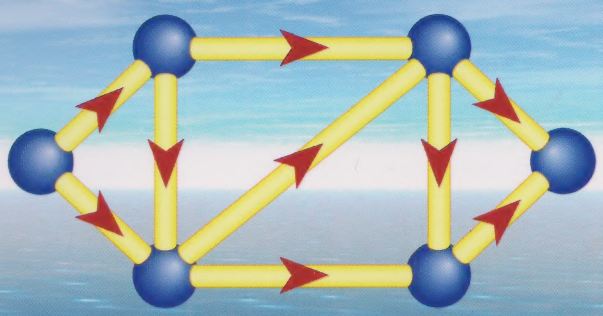
The Open University

MT365 Graphs, networks and design



Networks 1

Network flows



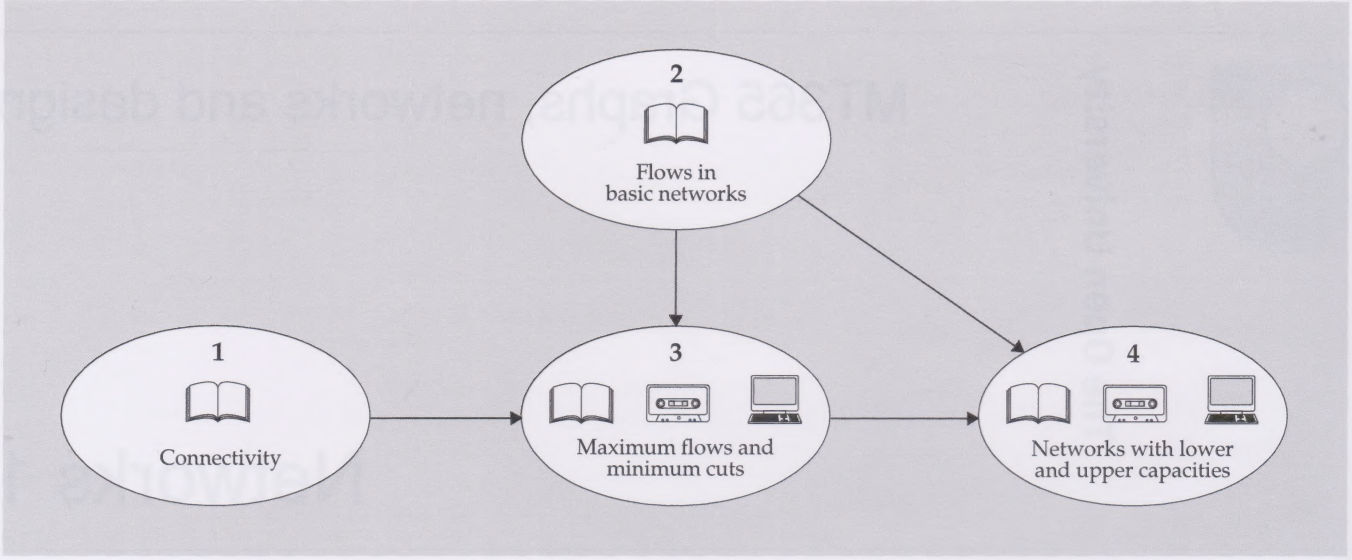
The Open University

MT365 Graphs, networks and design

Networks 1

Network flows

Study guide



Sections 1–3 introduce fundamental ideas in the study of networks, so it is important that you understand this material. Allow about one and a half evenings to study Section 1. Both Sections 3 and 4 include computer activities and audio-tape work. You will probably need to spend two evenings studying Section 3.

The Open University, Walton Hall, Milton Keynes, MK7 6AA.

First published 1995. Second edition 2009.

Printed and bound by Page Bros, Norwich.

ISBN 978 0 7492 5422 3

2.1

Contents

Introduction

Many problems of great importance concern the flow of things from one part of a system to another. For example, how much traffic can pass through a city in a given time? How frequently should a bus company run buses so as to cope with all the passengers? How much gas can flow through a pipeline in a given time interval? What electrical currents will flow through the components of a system? How does money flow throughout the world financial markets? How does information flow through organizations?

Some of these questions involve the maximum amount of a commodity (vehicles, passengers, oil, electricity, money) that can *flow* through a system in a given time. Others concern the *transmission* of effects through a system (wealth, traffic congestion).

In order for things to flow or be transmitted through a system, it is necessary for there to be a *channel* along which these things can pass. It is natural to represent these channels or links by *edges* in a graph or, more often, by *arcs* in a digraph when the direction of flow is relevant. It is then natural to represent any relevant information on the things that flow by *numbers* or *weights* on the edges or arcs to quantify the flow (number of vehicles in unit time, number of passengers carried, quantity of gas flowing in unit time, electrical current flowing, quantity of money flowing). In general, each channel or arc has a limitation on the flow it can carry, and this limit is called its *capacity*. Systems which can be represented in this way are examples of *networks*.

Networks are characterized by things flowing from one vertex to another along a sequence of intermediate arcs. In other words, things flow along *paths* in networks.

In order for things to flow between any vertices of a network, it is necessary that the network be *connected*. Recall that a (di)graph is (strongly) connected if there is a path from every vertex to every other. Intuitively, we would expect that the more paths there are between any two vertices, the easier it will be for things to flow between them. Similarly, we would expect that the more 'bottlenecks' there are in a network, the greater the obstruction to flow. These ideas are made precise later in this unit.

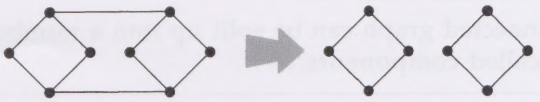
When designing new networks or managing existing networks, planners need to be able to answer a number of questions. These include the following.

- What happens if an arc in a network gets damaged — for example, an accident blocks a road, a bridge is closed for repairs, a pipe bursts, a computer goes down?
- What happens if a vertex of a network is lost — for example, an emergency closes a station, or a switchboard becomes jammed?
- Can the arcs of the network carry the flow required of them?
- Will the arcs of the network be able to carry a different pattern of flow in the future?
- If a network is unable to carry the predicted demand, can it be adapted to do so?
- If new arcs have to be added, which of the many possibilities should be chosen?

In order to plan systems for optimum flows and to be able to give sensible answers to such questions, it is necessary to understand how

connectivity properties can influence flows and dynamic changes in flow: network connectivity constrains network behaviour. In this unit we investigate the relation between connectivity and maximum flows through networks which have capacity restrictions on their arcs and/or vertices. We begin by showing how the intuitive concept of connectivity can be defined precisely. Somewhat paradoxically, this is done by considering how graphs and digraphs can be 'disconnected' by removing edges or arcs. For example, graph (b) below can be considered more connected than graph (a) because removal of just one edge disconnects graph (a), whereas removal of two edges is needed to disconnect graph (b).

(a) removal of one edge required to disconnect graph



(b) removal of two edges required to disconnect graph

The concept of removing certain edges (arcs) of a graph (digraph) ties up closely with the problem of determining the maximum flow that a network can carry. One of the main objectives of this unit is to prove the *max-flow min-cut theorem* which links these ideas explicitly and provides answers to many important practical questions.

We start, in Section 1, *Connectivity*, by explaining what it means for some graphs or digraphs to be 'more connected' than others. This leads to various forms of an important result on connectivity known as *Menger's theorem*. The section ends with a discussion of connectivity problems arising in telecommunications.

Next, in Section 2, *Flows in basic networks*, we introduce a simplified type of network, called a 'basic network', and look at ways of increasing the flow in such a network.

In Section 3, *Maximum flows and minimum cuts*, we introduce the idea of a *bottleneck* in a network, and we show how the maximum amount of flow in a network is related to the size of the smallest bottleneck restricting the flow. We also describe a method, called the *maximum flow algorithm*, which can be used to find a maximum flow and a minimum cut in any basic network, however large or complex. The material in this section is fundamental to any discussion of network flow problems.

In Section 4, *Networks with lower and upper capacities*, we consider networks with additional restrictions involving the *minimum* amount of flow through each arc, and we show how the basic ideas introduced in Section 3 can be generalized to deal with such networks.

1 Connectivity

In this section we investigate the extent to which a given graph or digraph is connected. In particular, we discuss the question: how many edges do we need to remove from a given connected graph so that it becomes disconnected? This, and other similar questions related to connectivity, are important ones to consider when designing telecommunication networks and road systems. For example, in a telecommunication network it is essential that the network should still be operable if some of the links between the exchanges become damaged, or are blocked by other calls. After discussing the theory of connectivity in graphs and digraphs in general, we describe its relevance to such networks.

1.1 Connected graphs and digraphs

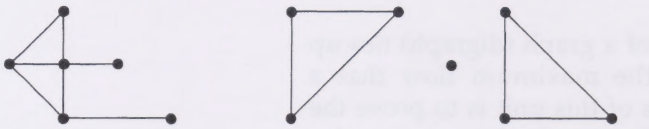
In *Graphs 1* we introduced the idea of a connected graph, that is, a graph which is 'in one piece'. We noted that in any connected graph there is a path between each pair of vertices, and this led to the following definitions.

Definitions

A graph is **connected** if there is a path between each pair of vertices, and is **disconnected** otherwise.

Every disconnected graph can be split up into a number of connected subgraphs, called **components**.

For example:



a connected graph

a disconnected graph with 3 components

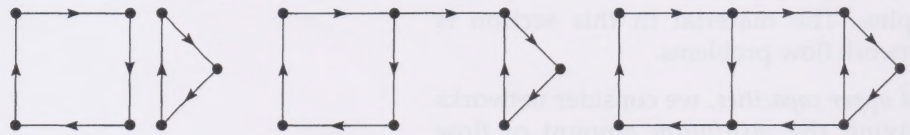
In the case of digraphs, it is not true in general that a digraph which is 'in one piece' must have a (directed) path between any given pair of vertices, and this observation led us to define two types of connected digraph, as follows.

Definitions

A digraph is **connected** if its underlying graph is connected, and is **disconnected** otherwise.

A digraph is **strongly connected** if there is a path from vertex u to vertex v for each pair of vertices u and v.

For example:



a disconnected digraph

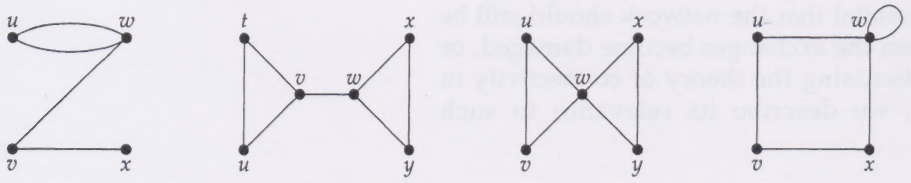
a connected digraph (not strongly connected)

a strongly connected digraph

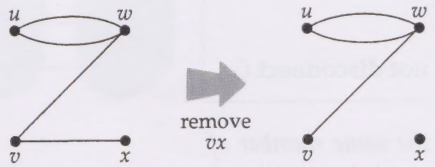
Edge connectivity

For many applications it is necessary to know more about a graph than just whether or not it is connected. For example, in telecommunication networks there are usually several different paths between a given pair of subscribers (vertices). In such a situation, it is important to know how many links (edges) can be blocked or broken without preventing a call being made between the two subscribers. In order to answer this and similar questions, we need to investigate connected graphs in more detail.

Consider the following graphs.

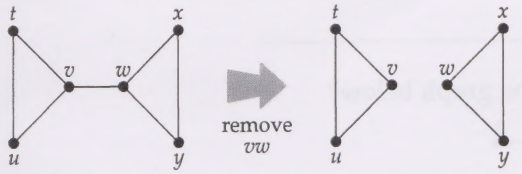


Graph (a) can be split into two components by removing either the edge vw or the edge vx. We say that the removal of either of these edges *disconnects* the graph.



(a)

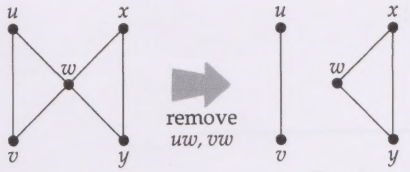
Graph (b) can also be disconnected by the removal of a single edge, the edge vw.



(b)

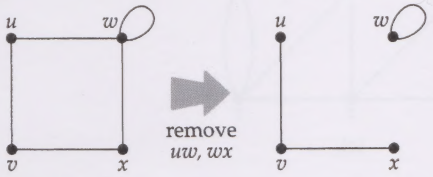
A single edge whose removal disconnects a graph, such as vw or vx in graph (a), or vw in graph (b), is called a **bridge**.

Graph (c) cannot be disconnected by removing a single edge, but the removal of two edges (such as uw and vw) disconnects it.



(c)

Similarly, graph (d) can be disconnected by removing two edges, for example, uw and wx.



(d)

With these examples in mind, we define the *edge connectivity* of a graph as follows.

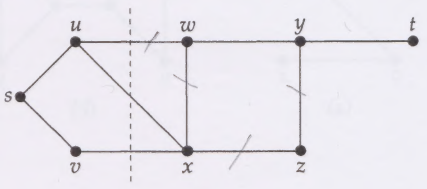
Definition

The **edge connectivity** $\lambda(G)$ of a connected graph G is the *smallest* number of edges whose removal disconnects G.

Thus graphs (a) and (b) have edge connectivity 1, and graphs (c) and (d) have edge connectivity 2.

If we wish to disconnect a graph by removing edges, we often have a choice of edges to delete. In view of this, it seems natural to consider ways of disconnecting a graph which do not involve the removal of 'redundant' edges.

Consider the graph G in the margin.



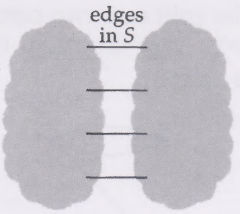
We can disconnect G by removing the three edges uw, ux and vx, but we cannot disconnect it by removing just two of these edges. We can also disconnect G by removing the four edges uw, wx, xz and yz, but the edge yz is redundant here, since we need remove only the edges uw, wx and xz to disconnect G. A set of such edges in which no edge is redundant — such as $\{uw, ux, vx\}$, $\{wy, xz\}$, or $\{yt\}$ — is called a *cutset*.

Definition

A **cutset** of a connected graph G is a set S of edges with the properties:

(a) removal of all the edges in S disconnects G;

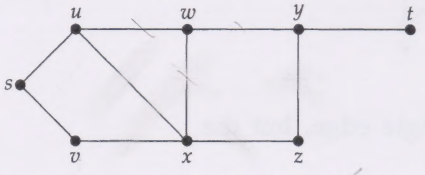
(b) removal of some (but not all) of the edges in S does not disconnect G.



Note that *two cutsets of a graph need not necessarily have the same number of edges*. For example, in the above graph, the sets $\{uw, ux, vx\}$, $\{wy, xz\}$ and $\{yt\}$ are all cutsets. Note also that *the edge connectivity* $\lambda(G)$ *of a graph* G *is the size of the* smallest *cutset of* G. For example, for the above graph, $\lambda(G) = 1$.

Problem 1.1

Which of the following sets of edges are cutsets of the graph below?

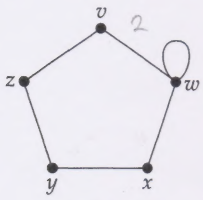


(a) $\{su, sv\}$; (b) $\{ux, wx, yz\}$; (c) $\{ux, vx, wx, yz\}$;

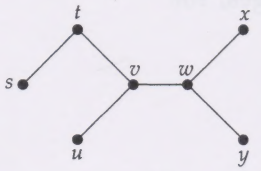
(d) $\{yt, yz\}$; (e) $\{wx, xz, yz\}$; (f) $\{uw, wx, wy\}$.

Problem 1.2

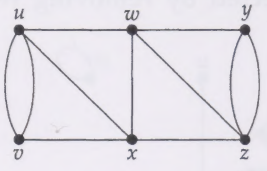
Write down the value of $\lambda(G)$ for each of the following graphs G.



(a)



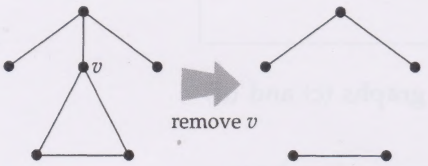
(b)



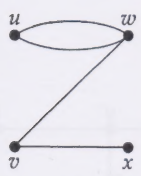
(c)

Vertex connectivity

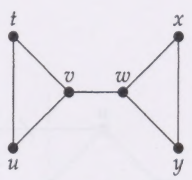
We can also think of connectivity in terms of the minimum number of *vertices* which need to be removed in order to disconnect a graph. When we remove a vertex, we must also remove the edges incident with it:



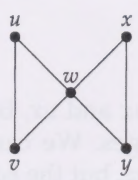
Consider again the following graphs.



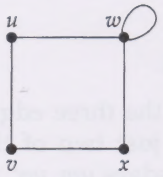
(a)



(b)



(c)



(d)

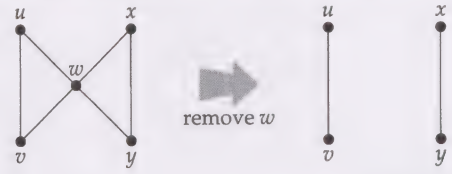
Graphs (a) and (b) can be disconnected by the removal of a single vertex (either v or w).

(a)

(b)

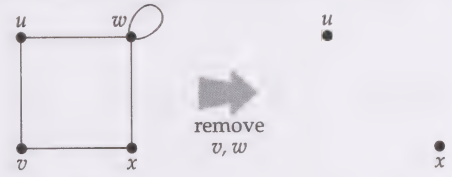
Graph (c) can also be disconnected by removing just one vertex, w.

A single vertex whose removal disconnects a graph, such as v or w in graph (a) or graph (b), or w in graph (c), is called a **cut vertex**.



(c)

Graph (d) cannot be disconnected by removing a single vertex, but the removal of two non-adjacent vertices (such as v and w) disconnects it.



(d)

With these examples in mind, we define the *connectivity* (or *vertex connectivity*) of a graph as follows.

Definition

The **connectivity** (or **vertex connectivity**) $\kappa(G)$ of a connected graph G (other than a complete graph) is the *smallest* number of vertices whose removal disconnects G.

We use the simpler term *connectivity* when there is no possibility of confusion with *edge connectivity*.

Thus graphs (a), (b) and (c) have connectivity 1, and graph (d) has connectivity 2.

The above definition breaks down if G is a complete graph, since we cannot then disconnect G by removing vertices. We therefore make the following definition.

Recall from *Graphs 1* that a complete graph is a graph in which each pair of distinct vertices is joined by exactly one edge. The complete graph with n vertices is denoted by K_n.

Definition

The **connectivity** $\kappa(K_n)$ of the complete graph K_n ($n \geq 3$) is $n - 1$.

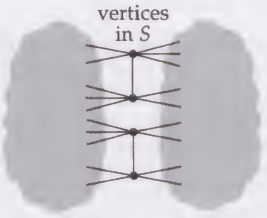
There is also a 'vertex analogue' of the concept of a cutset.

Definition

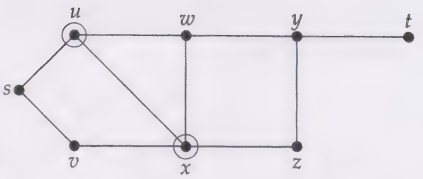
A **vertex cutset** of a connected graph G is a set S of vertices (not the whole set of vertices) with the properties:

(a) removal of all the vertices in S disconnects G;

(b) removal of some (but not all) of the vertices in S does not disconnect G.



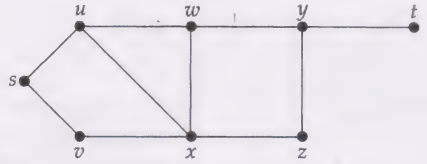
For example, we can disconnect the graph in the margin by removing the two vertices u and x, but we cannot disconnect it by removing just one of these vertices. It follows that $\{u, x\}$ is a vertex cutset.



Note that *two vertex cutsets of a graph need not necessarily have the same number of vertices.* For example, in the graph in the margin, the sets $\{u, x\}$ and $\{y\}$ are both vertex cutsets. Note also that the *vertex connectivity* $\kappa(G)$ *of a graph* G *is the size of the* smallest *vertex cutset of* G. For example, for the graph in the margin, $\kappa(G) = 1$.

Problem 1.3

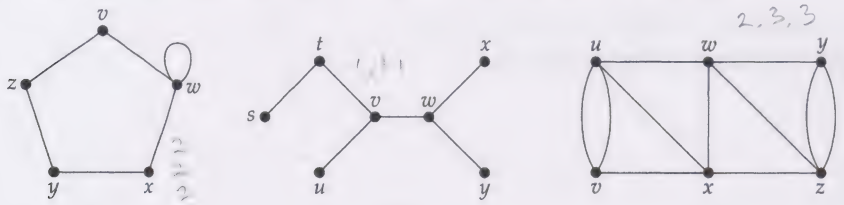
Which of the following sets of vertices are vertex cutsets of the graph below?



(a) $\{u, v\}$; (b) $\{v, w\}$; (c) $\{u, x, y\}$; (d) $\{w, z\}$.

Problem 1.4

Write down the value of $\kappa(G)$ for each of the following graphs G.



(a) (b) (c)

For each of these graphs, list the values of $\kappa(G)$, $\lambda(G)$ (found in Problem 1.2) and the smallest vertex degree $\delta(G)$.

In each of the above examples, you may have noticed that the vertex connectivity $\kappa(G)$ does not exceed the edge connectivity $\lambda(G)$, which does not exceed $\delta(G)$. This inequality holds for all connected graphs.

Theorem 1.1

For any connected graph G,

$$\kappa(G) \leq \lambda(G) \leq \delta(G),$$

where $\delta(G)$ is the smallest vertex degree in G.

Proof

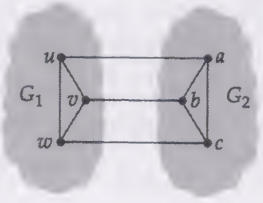
If G is the complete graph K_n, then $\kappa(G) = \lambda(G) = \delta(G) = n - 1$.

If G is not K_n, and if v is a vertex of degree $\delta(G)$, then G can be disconnected by removing all the $\delta(G)$ edges incident with v. It follows that $\lambda(G)$, the *minimum* number of edges whose removal disconnects G, cannot exceed $\delta(G)$. So $\lambda(G) \leq \delta(G)$.

It remains to show that $\kappa(G) \leq \lambda(G)$ whenever G is not a complete graph. It is sufficient to give a proof for the case when G is a simple graph.

Let G be a simple graph with n vertices and edge connectivity $\lambda(G)$. As G is not K_n, there is a vertex of degree at most $n - 2$, so, by what we have already proved, $\lambda(G) \leq n - 2$. There is at least one set of $\lambda(G)$ edges whose removal disconnects G into two components G_1 and G_2, as illustrated by the diagram in the margin. But we can also remove these edges by removing at

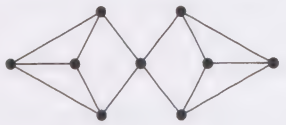
If we have a non-simple graph G and we can show that $\kappa(G') \leq \lambda(G')$ for the simple graph G' obtained by deleting loops and changing multiple edges to single edges, then $\kappa(G) \leq \lambda(G)$, since the changes cannot affect $\kappa(G)$ and can only decrease $\lambda(G)$, that is, $\kappa(G) = \kappa(G') \leq \lambda(G') \leq \lambda(G)$.



most $\lambda(G)$ vertices, since we have only to remove one suitably chosen end vertex from each of these $\lambda(G)$ edges. Let us remove these $\lambda(G)$ vertices one at a time. Since there are at most $n - 2$ edges to be removed, and since at each step we can remove a vertex either from G_1 or from G_2, we can achieve this by removing a set of vertices that leaves neither G_1 nor G_2 empty. It follows that the minimum number of vertices whose removal disconnects G cannot exceed $\lambda(G)$, that is, $\kappa(G) \leq \lambda(G)$. ■

Note that it is possible for both inequalities in Theorem 1.1 to be strict inequalities, that is, $\kappa(G) < \lambda(G) < \delta(G)$. For example, in the graph in the margin, $\kappa(G) = 1$, $\lambda(G) = 2$ and $\delta(G) = 3$.

For example, we can disconnect the graph above by removing the edges ua, vb and wc, or by removing the vertices u, v and c.



1.2 Menger's theorem for graphs

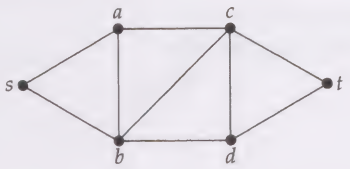
In this subsection we discuss an important result which relates the above ideas to the number of 'disjoint paths' between two vertices in a graph. This result is known as *Menger's theorem*.

We start by defining *disjoint paths* in a graph.

Definitions

Let G be a connected graph, and let s and t be vertices of G. A path between s and t is called an ***st*-path**. Two or more st-paths are **edge-disjoint** if they have no edges in common, and **vertex-disjoint** if they have no vertices in common (apart from s and t).

For example, in the following graph,



the paths *sact* and *sbdt* are both edge-disjoint and vertex-disjoint st-paths;

the paths *sact* and *sbct* are neither edge-disjoint nor vertex-disjoint (since they have the edge *ct* in common);

the paths *sact* and *sbcdt* are edge-disjoint, but not vertex-disjoint (since they have the vertex *c* in common).

Problem 1.5

Consider the graph in the margin.

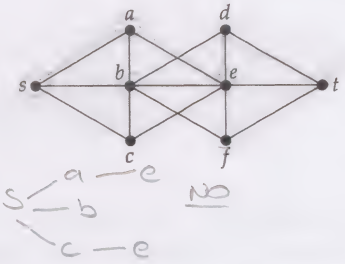
Write down

(a) three edge-disjoint st-paths;

(b) two st-paths which are edge-disjoint, but not vertex-disjoint;

(c) two vertex-disjoint st-paths.

Does this graph contain three vertex-disjoint st-paths? Justify your answer.



Problem 1.6

(a) Prove that if two st-paths in a graph are vertex-disjoint, then they must also be edge-disjoint.

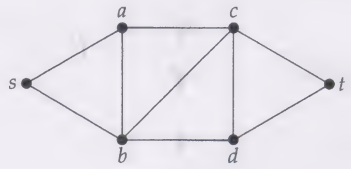
(b) Give an example of a graph in which no two edge-disjoint st-paths are vertex-disjoint.

We also need the following definitions.

Definitions

Let G be a connected graph, and let s and t be vertices of G. Then certain *edges* **separate s from t** if the removal of these edges destroys all paths between s and t. Similarly, certain *vertices* (excluding s and t) **separate s from t** if the removal of these vertices destroys all paths between s and t.

For example, in the following graph,

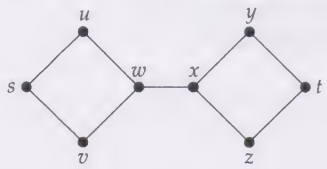


the edges ac, bc and bd separate s from t, as do the edges sa, ac, bc and bd;

the vertices b and c separate s from t, as do the vertices a, b and d.

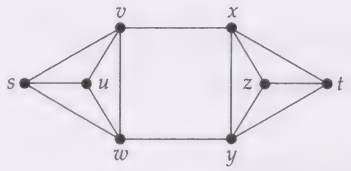
We now show how these ideas are related to those of edge-disjoint and vertex-disjoint st-paths.

Example 1.1



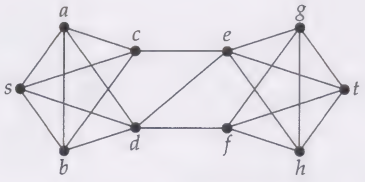
In this graph, the single edge wx separates s from t. It follows that *there cannot be two edge-disjoint st-paths*, since each st-path must include the edge wx. ■

Example 1.2



In this graph, the two edges vx and wy separate s from t. It follows that *there are at most two edge-disjoint st-paths*, since each st-path must include one of these two edges. ■

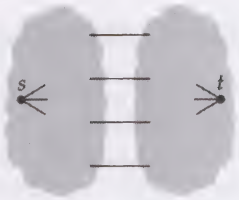
Example 1.3



In this graph, the three edges ce, de and df separate s from t. It follows that *there are at most three edge-disjoint st-paths*, since each st-path must include one of these three edges. ■

More generally, consider a set of edges separating s from t in an arbitrary connected graph. Since the removal of these edges destroys all paths

between s and t, each st-path must include at least one of them. It follows that *the maximum number of edge-disjoint st-paths cannot exceed the number of edges in this set.* Since this applies to *any* set of edges separating s from t, we have



the *maximum* number of edge-disjoint st-paths $\leq$ the number of edges in *any* set of edges separating s from t.

But this is true for *any* set of edges separating s from t, and so it must be true for *a set with the smallest possible number of edges.* So

the *maximum* number of edge-disjoint st-paths $\leq$ the *minimum* number of edges separating s from t.

The two numbers in the above inequality are, in fact, always equal. This is the *edge form* of Menger's theorem for graphs, which may be stated formally as follows.

Theorem 1.2: Menger's theorem for graphs (edge form)

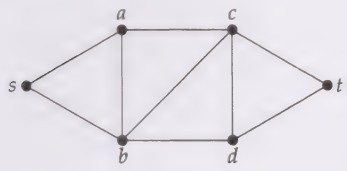
Let G be a connected graph, and let s and t be vertices of G. Then the maximum number of edge-disjoint st-paths is equal to the minimum number of edges separating s from t.

We outline the proof of this theorem in Section 3.4.

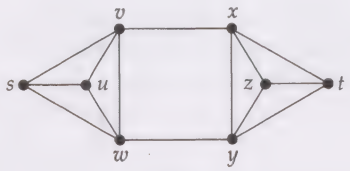
It follows from Menger's theorem that, if we can find k edge-disjoint st-paths and k edges separating s from t (for the same value of k), then k is the *maximum* number of edge-disjoint st-paths and the *minimum* number of edges separating s from t. Note that these k edges separating s from t necessarily form a cutset; it follows that, when looking for them, we need consider only cutsets whose removal disconnects G into two components, one containing s and the other containing t.

Problem 1.7

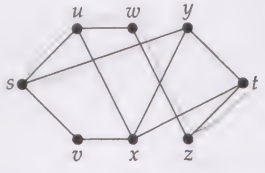
By finding k edge-disjoint st-paths and k edges separating s from t (for the same value of k), and by using the edge form of Menger's theorem, find the maximum number of edge-disjoint st-paths for each of the following graphs.



(a)



(b)



(c)

We can use Menger's theorem to obtain a result about edge connectivity. Recall that the edge connectivity $\lambda(G)$ of a connected graph G is the smallest number of edges whose removal disconnects G. So, by Menger's theorem, there are at least $\lambda(G)$ edge-disjoint paths between any given pair of vertices. We restate this result as follows.

Corollary of Menger's theorem for graphs (edge form)

A connected graph G has edge connectivity l if and only if every pair of vertices in G is joined by l or more edge-disjoint paths, and at least one pair of vertices is joined by exactly l edge-disjoint paths.

We shall find this corollary useful when we discuss telecommunication networks in Section 1.4.

1.3 Some analogues of Menger's theorem

We now present some analogues of Menger's theorem, starting with Menger's theorem for digraphs (arc form), and continuing with the vertex forms for both graphs and digraphs. *At this stage we state all these results without proof.* In Section 3 we show how Menger's theorem for digraphs (arc form) can be deduced from a theorem about network flows called the *max-flow min-cut theorem*, and we indicate how other variations of Menger's theorem can be deduced.

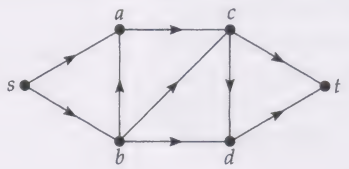
Menger's theorem for digraphs (arc form)

Many of the concepts introduced earlier for graphs have analogues for digraphs. For example, the following definitions are almost identical to those given for graphs.

Definitions

Let D be a connected digraph, and let s and t be vertices of D. A path from s to t is called an ***st*-path**. Two or more st-paths are **arc-disjoint** if they have no arcs in common, and **vertex-disjoint** if they have no vertices in common (apart from s and t).

For example, in the following digraph,



the paths *sact* and *sbdt* are both arc-disjoint and vertex-disjoint st-paths;
the paths *sact* and *sbct* are neither arc-disjoint nor vertex-disjoint;
the paths *sact* and *sbcdt* are arc-disjoint but not vertex-disjoint.

Definitions

Let D be a connected digraph, and let s and t be vertices of D. Then certain *arcs* **separate *s* from *t*** if the removal of these arcs destroys all paths from s to t. Similarly, certain *vertices* (excluding s and t) **separate *s* from *t*** if the removal of these vertices destroys all paths from s to t.

For example, in the above digraph:
the arcs *ac*, *bc* and *bd* separate s from t, as do the arcs *sa*, *ac*, *bc*, *bd* and *dt*;
the vertices *b* and *c* separate s from t, as do the vertices *a*, *b* and *d*.

We now restate the arc form of Menger's theorem for digraphs.

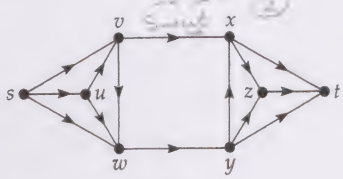
Theorem 1.3: Menger's theorem for digraphs (arc form)

Let D be a connected digraph, and let s and t be vertices of D. Then the maximum number of arc-disjoint st-paths is equal to the minimum number of arcs separating s from t.

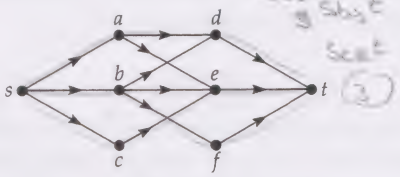
As with Menger's theorem for graphs, if we can find k arc-disjoint st-paths and k arcs separating s from t (for the same value of k), then k is the *maximum* number of arc-disjoint st-paths and the *minimum* number of arcs separating s from t.

Problem 1.8

By finding k arc-disjoint st-paths and k arcs separating s from t (for the same value of k), and by using the arc form of Menger's theorem, find the maximum number of arc-disjoint st-paths for each of the following digraphs.



(a)



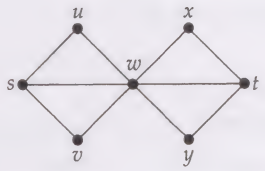
(b)

Menger's theorem for graphs (vertex form)

We have seen how Menger's theorem (edge form) relates the number of edge-disjoint st-paths in a graph to the smallest number of edges separating s from t, and how this result relates to edge connectivity. We shall state an analogous theorem for vertex-disjoint st-paths, but first we look at some examples.

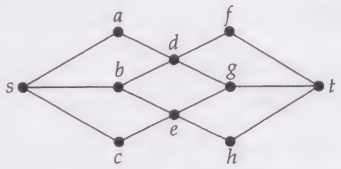
This is the original version of Menger's theorem, proved by K. Menger in 1927. The corollary was proved five years later by H. Whitney. The edge form and arc form of Menger's theorem were proved in 1955 by L. R. Ford and D. R. Fulkerson.

Example 1.4



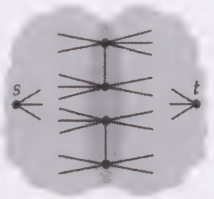
This graph has (vertex) connectivity 1, and the vertex w separates s from t. It follows that *there cannot be two vertex-disjoint st-paths*, since each st-path must include the vertex w. ■

Example 1.5



This graph has connectivity 2, and the vertices d and e separate s from t. It follows that *there are at most two vertex-disjoint st-paths*, since all st-paths must include one of these vertices. ■

More generally, consider a set of vertices (excluding s and t) separating non-adjacent vertices s and t in an arbitrary connected graph. Since the removal of these vertices destroys all paths between s and t, each st-path must include at least one of them. It follows that *the maximum number of vertex-disjoint st-paths cannot exceed the number of vertices in this set.*



As with the edge form of Menger's theorem, these numbers are, in fact, equal. This is the *vertex form* of Menger's theorem, which we state formally as follows.

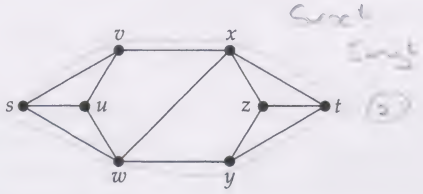
Theorem 1.4: Menger's theorem for graphs (vertex form)

Let G be a connected graph, and let s and t be non-adjacent vertices of G. Then the maximum number of vertex-disjoint st-paths is equal to the minimum number of vertices separating s from t.

As before, it follows that, if we can find k vertex-disjoint st-paths and k vertices separating s from t (for the same value of k), then k is the *maximum* number of vertex-disjoint st-paths and the *minimum* number of vertices separating s from t. Note that these k vertices separating s from t necessarily form a vertex cutset; it follows that, when looking for them, we need consider only vertex cutsets whose removal disconnects G into two or more components, one containing s and another containing t.

Problem 1.9

By finding k vertex-disjoint st-paths and k vertices separating s from t (for the same value of k), and by using the vertex form of Menger's theorem, find the maximum number of vertex-disjoint st-paths for the following graph.



We can also use this theorem to obtain a result about vertex connectivity. Recall that the connectivity $\kappa(G)$ of a graph G (other than a complete graph) is the smallest number of vertices whose removal disconnects G. So, by the vertex form of Menger's theorem, we can find a set of at least $\kappa(G)$ vertex-disjoint paths between any given pair of vertices. We can restate this result as follows.

Corollary of Menger's theorem for graphs (vertex form)

A connected graph G (other than a complete graph) has vertex connectivity k if and only if every non-adjacent pair of vertices in G is joined by k or more vertex-disjoint paths, and at least one non-adjacent pair of vertices is joined by exactly k vertex-disjoint paths.

We shall find this corollary useful when we discuss telecommunication networks in Section 1.4.

Menger's theorem for digraphs (vertex form)

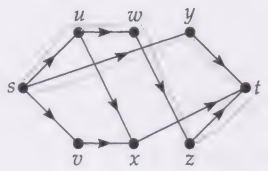
Finally, for completeness, we present the vertex form of Menger's theorem for digraphs. This is almost identical to the vertex form for graphs.

Theorem 1.5: Menger's theorem for digraphs (vertex form)

Let D be a connected digraph, and let s and t be non-adjacent vertices of D. Then the maximum number of vertex-disjoint st-paths is equal to the minimum number of vertices separating s from t.

Problem 1.10

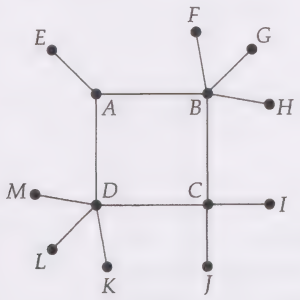
By finding k vertex-disjoint st-paths and k vertices separating s from t (for the same value of k), and by using the vertex form of Menger's theorem, find the maximum number of vertex-disjoint st-paths for the following digraph.



1.4 Reliable telecommunication networks

In this section we look at the use of graph theory to represent telecommunication networks and show how it can be of value in the design of efficient networks. We shall see that the notion of connectivity is important in this context.

A graph of a telecommunication network may contain a very large number of vertices and edges. The vertices may represent telephone exchanges and subscribers, or they may represent the switches in an exchange. In each case the edges represent links between them. It is important that such a network should be reliable. One aspect of reliability is that calls between subscribers should be possible even if a few exchanges or links are damaged. Consider the following graph, which represents a possible interconnection of exchanges.



If the link AB is damaged, then communication between exchanges A and B is still possible, but if the link EA is damaged, then exchange E cannot communicate with any other exchange. The minimum number of links which, when damaged, prevent the system from functioning fully is equal to the *edge connectivity* of the corresponding graph. Similarly, the *vertex connectivity* tells us how many exchanges must suffer damage before there is a breakdown in communications between the remaining exchanges.

So one reason for providing alternative paths between exchanges (or other types of vertex) is so that communication between the exchanges is still possible if one path is damaged. Another important reason for providing alternative paths is that a particular link or exchange in the path between two exchanges may also form part of the path between another pair of exchanges. The link or the exchange may therefore be already used to capacity when a new call is attempted, thus preventing the new call from being made. In such a case we say that the new call is *blocked*.

For two particular exchanges, the maximum number of alternative paths, no two of which pass through the same intermediate exchange, is (in the terminology of graph theory) the maximum number of vertex-disjoint paths between the corresponding vertices of the graph. By the corollary to the vertex form of Menger's theorem, the smallest number of such paths between the two exchanges is equal to the *vertex connectivity*. Similarly, the maximum number of alternative paths, no two of which use the same link between exchanges, is the maximum number of edge-disjoint paths between the corresponding vertices of the graph. By the corollary to the edge form of Menger's theorem, the smallest number of such paths between any two exchanges is equal to the *edge connectivity*.

If reliability were the only consideration, a telecommunication system would have as many alternative paths as possible between exchanges, that is, it would have the largest possible vertex connectivity and the largest possible edge connectivity. To achieve this, each exchange would need to be connected to every other exchange, and the corresponding graph would be a complete graph. This is clearly impracticable. What can be done, however, is to try to achieve the largest possible values of the vertex connectivity $\kappa(G)$ and the edge connectivity $\lambda(G)$ for a graph G with a given number of vertices and edges.

We know from Theorem 1.1 that $\kappa(G) \leq \lambda(G) \leq \delta(G)$ for any connected graph G, where $\delta(G)$ is the smallest vertex degree in G. Suppose G has n vertices and m edges. Then, by the handshaking lemma, the sum of all the vertex degrees is $2m$. It follows that the *average* of the vertex degrees is $2m/n$, so the *minimum* degree of the vertices cannot be greater than this. Combining these results, we obtain the inequalities:

The handshaking lemma was introduced in *Graphs 1*, Section 1.

$$\kappa(G) \leq \lambda(G) \leq \delta(G) \leq 2m/n.$$

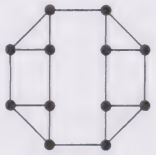
A graph G for which these inequalities are all equalities has the maximum vertex connectivity and the maximum edge connectivity possible for any graph with n vertices and m edges. Such a graph is said to have **optimal connectivity**. To show that a graph G has optimal connectivity, it is sufficient to show that $\kappa(G) = 2m/n$, as then the above inequalities guarantee that $\kappa(G) = \lambda(G) = \delta(G)$.

Definition

Let G be a graph with n vertices and m edges. Then G has **optimal connectivity** if

$$\kappa(G) = 2m/n.$$

All graphs with optimal connectivity are regular graphs (since the smallest vertex degree equals the average of the vertex degrees), but not every regular graph has optimal connectivity. For example, the graph G in the margin is regular of degree 3, so $\delta(G) = 3$; but $\kappa(G) = \lambda(G) = 2$, so G does not have optimal connectivity.



Problem 1.11

Show that the following regular graphs all have optimal connectivity:

(a) the cycle graph C_n $(n \geq 3)$;

(b) the complete graph K_n $(n \geq 3)$;

(c) the complete bipartite graph $K_{r,r}$ $(r \geq 2)$.

Problem 1.12

(a) There are two non-isomorphic simple graphs with 6 vertices and 9 edges which have optimal connectivity. Draw them.

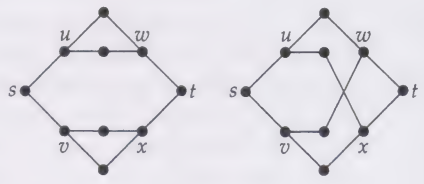
(b) Draw a regular simple graph with 7 vertices and 14 edges, which does not have optimal connectivity.

You may like to defer this problem and use the computer.

We have seen that the values of $\kappa(G)$ and $\lambda(G)$ for a graph G give us some information about the reliability of the corresponding telecommunication system, or part of such a system. However, these values do not tell us the whole story: two graphs with the same number of vertices, the same number of edges, and the same values of $\kappa(G)$ and $\lambda(G)$, may not correspond to equally reliable systems.

Consider, for example, the two graphs in the margin, each of which has 10 vertices and 12 edges, and satisfies $\kappa(G) = \lambda(G) = 2$.

In the first graph, all paths from the vertex s to the vertex t can be destroyed by removing the edges of any one of the four cutsets with two edges separating s from t — namely, $\{su, sv\}$, $\{wt, xt\}$, $\{su, xt\}$ or $\{sv, wt\}$. However, the second graph has only two cutsets with two edges separating s from t — namely, $\{su, sv\}$ and $\{wt, xt\}$. So if all the edges contained in these cutsets have equal likelihood of being blocked or damaged, we would expect the second graph to be more reliable than the first, and it can be shown that this is indeed the case.



In order to have full information about the reliability of a network represented by a graph, we need to know not only the edge connectivity and vertex connectivity, but also all the cutsets of the graph.

Although the account given here is necessarily simplified, the ideas presented in this section have proved to be of great importance in the design and analysis of telecommunication systems.

After studying this section, you should be able to:

- explain the terms *edge connectivity*, *vertex connectivity*, *cutset* and *vertex cutset*;
- understand and use the various forms of Menger's theorem;
- discuss the reliability of a telecommunication network in terms of the concepts introduced in this section.

2 Flows in basic networks

How much gas can flow through a given pipeline network in a day? How many vehicles can pass through a town in a given period of time? How can a manufacturer send the maximum number of items from factories to retailers in a given period of time? Each of these questions involves finding the maximum amount of a commodity (gas, traffic, manufactured items) that can pass through a network in a given time. In each case, there are restrictions (such as pipe diameter, road width, cost) limiting the amount of flow through the channels of the network. From now on, we are concerned with the problem of finding these 'maximum flows'.

2.1 Basic networks

We start by restricting our attention to a simplified type of network.

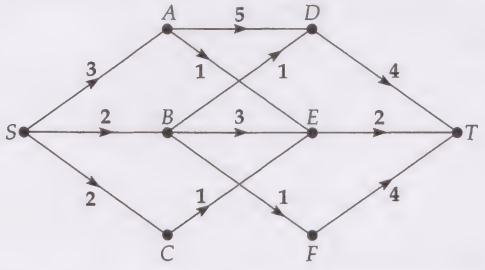
Definitions

A **basic network** is a connected digraph N satisfying the conditions:

(a) N has exactly one **source** and one **sink**;

(b) each arc e of N is assigned a positive number $c(e)$, called the **capacity of e**, which represents the maximum amount of the commodity that can flow through the arc in a given time.

A typical example of a basic network is the following; the number next to an arc is the capacity of the arc.



We usually use the letters S and T to signify the *source* and the *sink*, respectively.

We now formally define a *flow* in such a network, that is, a description of the amount of a commodity that can flow along the various channels of the network in a given period of time. We must ensure that no channel receives more of the commodity than it can cope with, and that none of the commodity is lost along the way. We make the following definition.

Definition

A **flow** in a basic network N with source S and sink T is an assignment, to each arc e of N, of a non-negative number $f(e)$, called the **flow along the arc** e, satisfying:

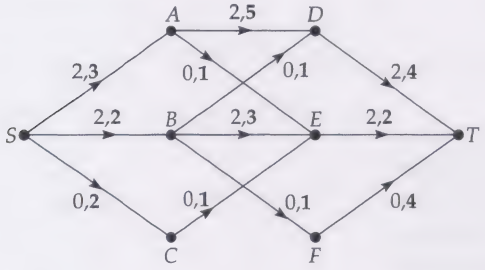
(a) the **feasibility condition**: the flow along each arc does not exceed the capacity of the arc, that is,

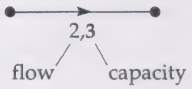
flow ≤ capacity;

(b) the **flow conservation law**: for each vertex V other than S and T, the sum of the flows along the arcs into V is equal to the sum of the flows along the arcs out of V.

Such a flow is sometimes called a *flow from S to T* or an *(S,T)-flow*.

An example of a flow in a basic network is shown below; the first number next to each arc e is the flow $f(e)$, and the second number (in bold type) is the capacity $c(e)$.





An arc which can take no further flow — such as SB and ET above — is said to be *saturated*.

Definition

An arc e is **saturated** if $f(e) = c(e)$, and **unsaturated** if $f(e) < c(e)$.

Problem 2.1

This problem refers to the above network.

(a) Check that the flow satisfies the feasibility condition and the flow conservation law.

(b) Verify that the total flow out of S is equal to the total flow into T. Explain why this must always be the case.

By the flow conservation law, none of the flow is lost at any vertex, and so the total flow out of the source S is always equal to the total flow into the sink T. This common value is called the **value of the flow**, and represents the amount of the commodity flowing through the network. Since our primary concern is with the *maximum* amount of the commodity flowing from S to T, it is natural to make the following definition.

In the example given above, the value of the flow is 2 + 2 + 0 = 4.

Definition

A **maximum flow** is a flow of largest possible value.

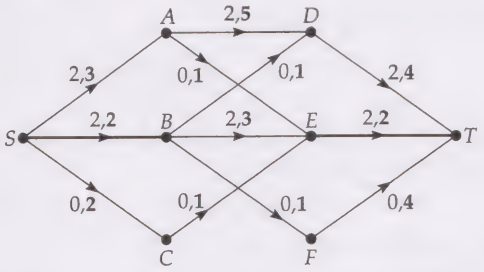
Note that we say *'a* flow of largest possible value' because there may be several different flows with the same largest value.

In the next section we consider the problem of finding a maximum flow in a given basic network.

2.2 Flow-augmenting paths

To find a maximum flow, we first find an initial flow by inspection. We then increase the value of the flow step by step until we can increase it no further. Finding an initial flow is not difficult, since we can always take the *zero flow* in which the flow in each arc is 0.

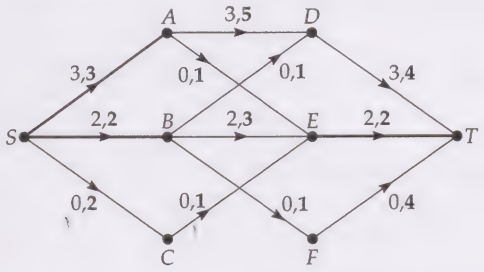
To illustrate the procedure, we return to our example above. Here we can find a non-zero flow by inspection: a flow of value 2 along $SADT$ and a flow of value 2 along $SBET$, giving a flow from S to T of value 4.



We have represented the saturated arcs by thick lines. By doing this, we can see at a glance which arcs are unsaturated and so can take a larger flow.

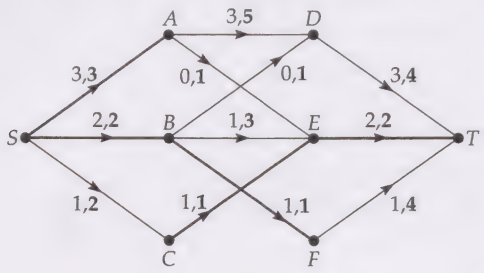
A path from S to T consisting entirely of unsaturated arcs is an example of a *flow-augmenting path*. By increasing the flow along the arcs of such a path, we can increase the value of the flow.

For example, the path $SADT$ consists entirely of unsaturated arcs, and we can increase the flow along each of its arcs by 1 (so that SA becomes saturated with a flow of 3, AD has a flow of 3, and DT has a flow of 3); this increases the value of the flow from 4 to 5:



There are now no paths from S to T consisting entirely of unsaturated arcs (since any such path must start with SCE, and there are no unsaturated arcs out of E), and so the process cannot be repeated.

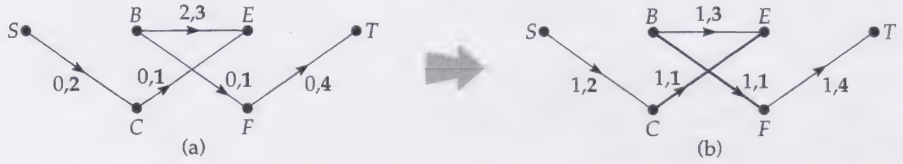
However, the flow of value 5 is not a maximum flow. The following diagram illustrates a flow of value 6.



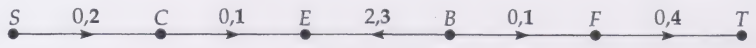
Since our primary aim is to construct maximum flows, this is rather discouraging. Either we must abandon the method or we must find some way of modifying it so that it works in all cases.

Fortunately the situation is not as bad as it seems. All we need to do is to specify more carefully what we mean by a 'flow-augmenting path'. Up to now this term has meant a path from S to T consisting entirely of

unsaturated arcs, but we now give it a less restricted meaning. To see what is involved, compare the above two diagrams. To get the flow with value 6 from the one with value 5 we *increase* the flow by 1 along *SC*, *CE*, *BF* and *FT*, and *decrease* the flow by 1 along *BE*:

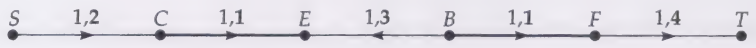


We can think of diagram (a) as a 'path' containing four *forward* arcs *SC*, *CE*, *BF* and *FT*, and one *backward* arc *EB*:



Note that we write the backward arc as *EB*, and not *BE*, since the flow is directed from *S* to *T*.

To obtain diagram (b), we *increase* the flow by 1 along the forward arcs of this path and *decrease* the flow by 1 along the backward arc:



This is possible, since the forward arcs are all unsaturated (so that the flow along them can be increased) and the backward arc carries a non-zero flow (which can be decreased). This idea of increasing the flow along forward arcs and decreasing the flow along backward arcs leads to the following definitions.

Definitions

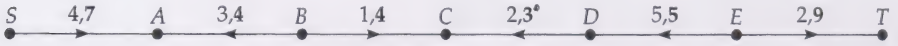
A **flow-augmenting path** in a basic network with source *S* and sink *T* is a path from *S* to *T* consisting of:

forward arcs: unsaturated arcs directed along the path

and, possibly,

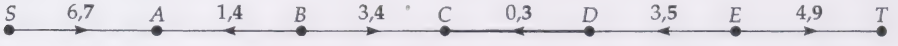
backward arcs: arcs directed against the direction of the path and carrying a non-zero flow.

Another example of a flow-augmenting path in a basic network is:



The arcs *SA*, *BC* and *ET* are forward arcs, and the arcs *AB*, *CD* and *DE* are backward arcs.

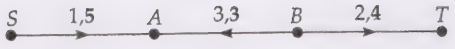
We can increase the value of the flow from *S* to *T* by 2, by increasing the flow along the forward arcs by 2 and decreasing the flow along the backward arcs by 2:

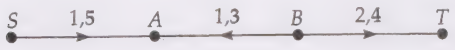


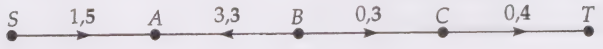
Since the backward arc *CD* now carries a zero flow (which cannot be decreased), we cannot increase the value of the flow any further. Here, we have indicated this by representing the arc *CD* by a thick line.

Problem 2.2

For each of the following flow-augmenting paths, find the largest amount by which the flow from *S* to *T* can be increased, and draw a diagram showing the resulting flow.

(a) 

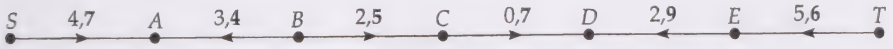
(b) 

(c) 

Problem 2.3

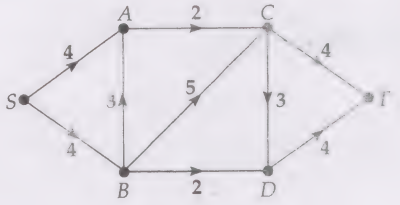
(a) Give a general rule for calculating the largest amount by which the flow can be increased along a flow-augmenting path.

(b) Verify that your rule works by applying it to the flow-augmenting path:



Problem 2.4

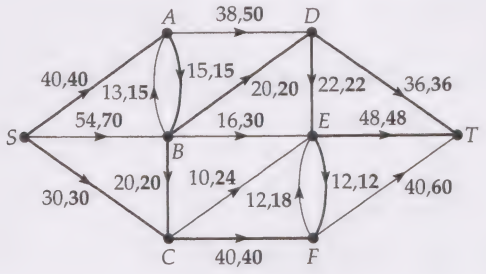
By finding flow-augmenting paths, find a maximum flow in the following basic network; the number next to an arc is the capacity of the arc.



For small networks, we can be fairly confident when we have found a maximum flow. However, for large networks, it may not be obvious when a maximum flow has been attained. The following problem demonstrates the need for a more systematic method of finding a maximum flow.

Problem 2.5

The following diagram shows a network with a flow of value 124. Is this a maximum flow?



2.3 Transforming to a basic network

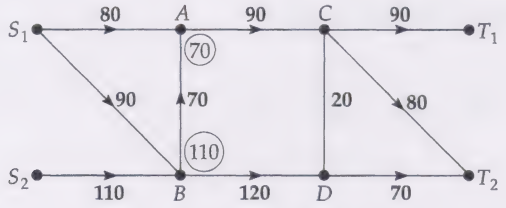
In Sections 2.1 and 2.2 we considered the flow problem for basic networks, and we described a way of constructing a maximum flow in such a network by finding flow-augmenting paths. However, many networks which arise in practice do not satisfy the three conditions which are satisfied by a basic network:

- all edges are directed;
- there are no capacity restrictions on the vertices;
- there is only one source and only one sink.

For example, the network of streets in a town does not usually satisfy these conditions, since it generally involves both one-way and two-way streets, has road junctions at which the traffic flow is restricted by traffic signals, and contains several points at which traffic can enter or leave the system. In this subsection we show how the ideas developed for basic networks can be adapted to find maximum flows in networks which do not satisfy the above conditions.

As an illustrative example, we consider the following network throughout this subsection.

Example 2.1



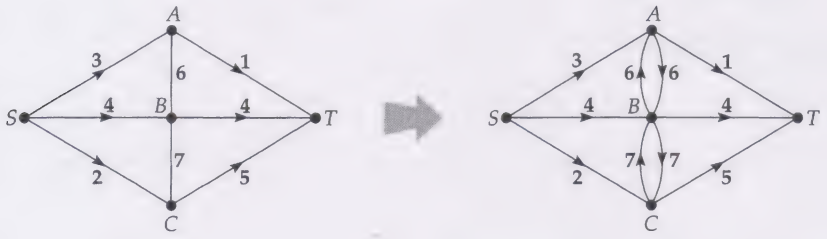
Here,

- the edge CD is undirected;
- each of the vertices A and B has a capacity restriction (indicated by a circled number next to the vertex);
- there are two sources, S_1 and S_2, and two sinks, T_1 and T_2. ■

Undirected and mixed networks

An **undirected network** is one in which flow is allowed in either direction along each edge. A network containing both directed and undirected edges (such as a road network with both one-way and two-way streets) is called a **mixed network**. We can transform a mixed or undirected network into a basic network by regarding each two-way channel (undirected edge) as two one-way channels (arcs), one in each direction. In other words, we replace each undirected edge in the network by two arcs with the same capacity:

For example, the following mixed network with two undirected edges gives rise to the basic network on the right.



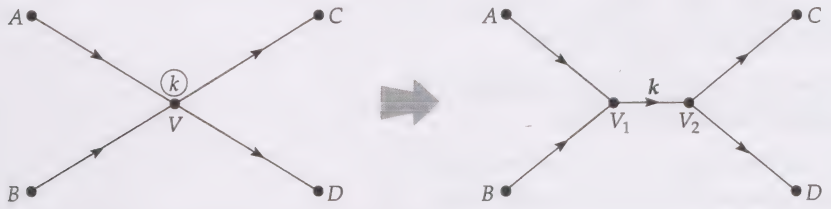
Problem 2.6

In the network of Example 2.1, replace the undirected edge CD by two arcs.

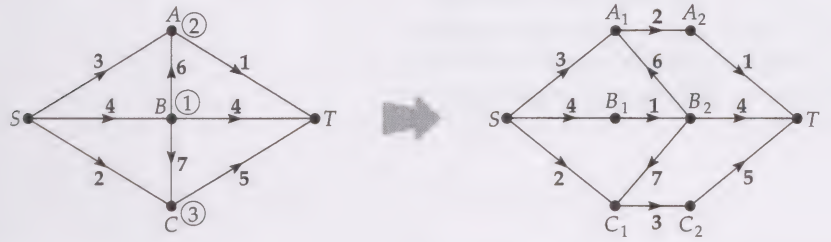
Networks with capacity restrictions on the vertices

If a network has one or more vertices with capacity restrictions (such as the flow of traffic through a particular junction, or the flow of mail through a particular sorting office), we can transform it into a basic network by replacing each such vertex by two vertices joined by an arc.

More precisely, we replace each vertex V of capacity k by two vertices V_1 and V_2 joined by an arc from V_1 to V_2 of capacity k. All arcs directed towards V in the original network are directed towards V_1 in the new network, and all arcs directed away from V in the original network are directed away from V_2 in the new network, as follows.



For example, the following network with capacity restrictions on the vertices gives rise to the basic network on the right.

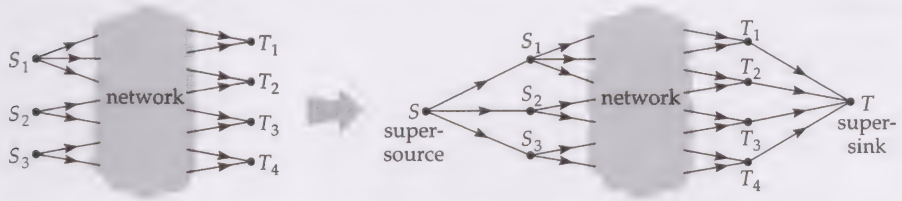


Problem 2.7

In the network given in the solution to Problem 2.6, replace the vertices A and B to obtain a network without capacity restrictions on the vertices.

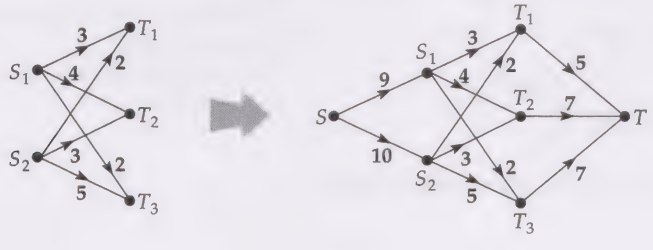
Networks with several sources and sinks

In many networks arising in practice there are a large number of sources and sinks corresponding to, for example, factories and markets in economic networks or telephone subscribers in telecommunication networks. If a network has several sources $S_1, S_2, \ldots$, and several sinks $T_1, T_2, \ldots$, we can transform it into a basic network by adjoining two new vertices — a *super-source* S joined to all the existing sources by new arcs, and a *super-sink* T joined to all the existing sinks by new arcs, as follows.



Each new arc SS_i is assigned a capacity equal to the sum of the capacities of the arcs out of S_i, and each new arc T_iT is assigned a capacity equal to the sum of the capacities of the arcs into T_i. Note that the value of the maximum flow from S to T in this basic network is equal to the value of the maximum flow from all the original sources to the original sinks.

For example, in the following network the arc SS_1 is given capacity $3 + 4 + 2 = 9$, and the arc T_3T is given capacity $2 + 5 = 7$.



Some authors assign infinite capacity to all the new arcs SS_i and T_iT to indicate that any amount of flow is allowed through these arcs. In fact, in our example any capacities can be assigned to the new arcs SS_i and T_iT, as long as they are not less than the given capacities.

Problem 2.8

In the network given in the solution to Problem 2.7, add a super-source S and a super-sink T, together with the necessary new arcs.

The transformations to convert a network into a basic network are summarized below, and should be carried out in the order given.

1 Replace each undirected edge by two arcs, one in each direction, both with the same capacity as the original undirected edge.

2 Replace each vertex V with a capacity restriction k by two vertices V_1 and V_2 joined by an arc from V_1 to V_2 of capacity k. All arcs directed towards V are directed towards V_1, and all arcs directed away from V are directed away from V_2.

3 If there are several sources $S_1, S_2, \ldots, S_i, \ldots$, join them to a new super-source S. If there are several sinks $T_1, T_2, \ldots, T_i, \ldots$, join them to a new super-sink T. To each new arc SS_i, assign a capacity equal to the sum of the capacities out of S_i; to each new arc T_iT assign a capacity equal to the sum of the capacities into T_i.

General procedure

We can now solve a variety of network flow problems by the following procedure.

STEP 1 Using the three transformations described above, transform the problem into one involving a basic network.

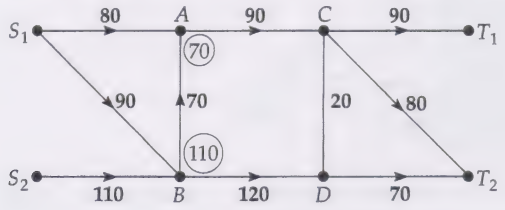
STEP 2 Solve this basic network problem.

STEP 3 Interpret the solution in terms of the original network.

The following worked problem illustrates this procedure.

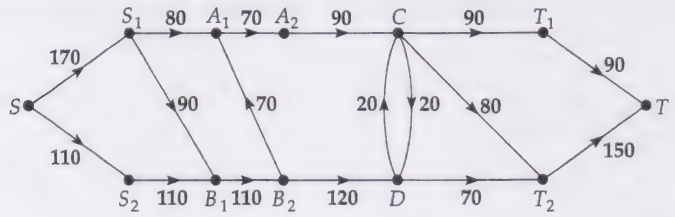
Worked problem

A manufacturer of farm machinery is faced with the problem of sending as many tractors as possible from two factories S_1 and S_2 to two markets T_1 and T_2 along the channels of the following network. The capacities of the arcs and the circled numbers on the vertices A and B are the numbers of tractors which can be transported in a week. How many tractors can be sent in a week from the factories to the markets?



Solution

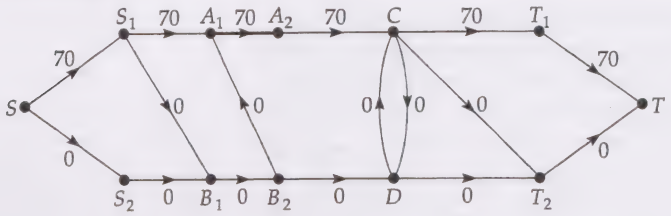
STEP 1 Applying the three transformations described above, we get the following basic network.



This is the network obtained in Problems 2.6–2.8.
Here the number next to an arc is the *capacity* of the arc.

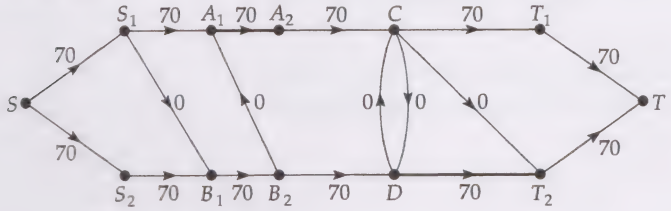
STEP 2 Using the method of Section 2.2, we can find a maximum flow in this basic network as follows.

We can send a flow of value 70 along the flow-augmenting path $SS_1A_1A_2CT_1T$:

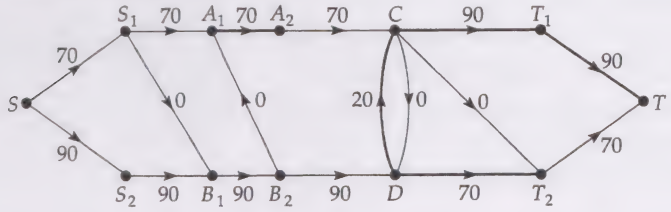


Here the number next to an arc is the *flow* along the arc.

We can send a flow of value 70 along the flow-augmenting path $SS_2B_1B_2DT_2T$:

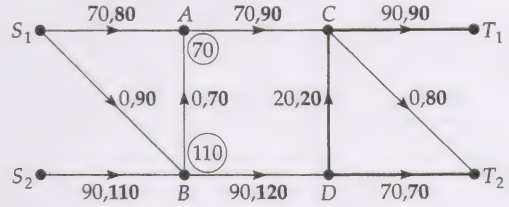


Finally, we can send a flow of value 20 along the flow-augmenting path $SS_2B_1B_2DCT_1T$:



No further increase in flow is possible, since the arcs A_1A_2, DC and DT_2 are all saturated. Thus this basic network has a maximum flow of value 160.

STEP 3 It follows that the manufacturer can send 160 tractors per week as indicated on the following diagram (obtained by 'undoing' the three transformations of Step 1).

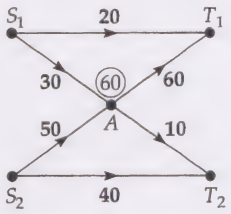


In 'undoing' Step 1 we have placed an arrow on the edge DC.

■

Problem 2.9

Use the above general procedure to find a maximum flow from the sources S_1 and S_2 to the sinks T_1 and T_2 for the network in the margin.



After studying this section, you should be able to:

- explain the terms *basic network, flow, saturated arc, value of a flow, maximum flow, forward arc, backward arc* and *flow-augmenting path*;
- find flow-augmenting paths in small basic networks by inspection;
- convert the following to basic networks:
  (a) undirected and mixed networks;
  (b) networks with capacity restrictions on the vertices;
  (c) networks with several sources and sinks.

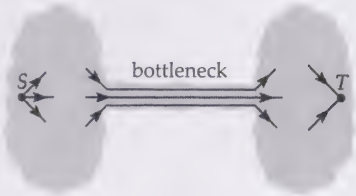
3 Maximum flows and minimum cuts

In the previous section we described a way of obtaining a maximum flow in a small basic network. In this method we identify flow-augmenting paths by inspection and build up the flow step by step until we obtain a maximum flow. However, this method is unsatisfactory for large networks in which the flow-augmenting paths are not obvious, and we have as yet no reliable way of knowing when we have found a maximum flow. In this section we describe an alternative method for determining whether a given flow is a maximum flow, and we describe an algorithm, called the *maximum flow algorithm*, for systematically identifying flow-augmenting paths and thus finding a maximum flow in a given network, however large or complicated. The algorithm also gives a *minimum cut*.

First, we introduce the idea of a *minimum cut* or *bottleneck* in a network, and we show how the maximum flow in a network is related to the size of the smallest bottleneck restricting the flow.

3.1 Minimum cuts

Consider a basic network with a bottleneck. In a road network this may result from an accident or roadworks, or in a pipeline network it can occur if the pipe diameter at some point becomes very small. In each case, the amount of flow through the network is limited by the amount of flow through the bottleneck.



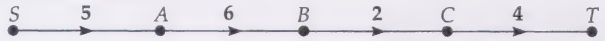
It seems that information on the amount of flow through the 'worst bottleneck' in the system may give information on the maximum amount of flow in the network. This is indeed the case, and forms the underlying idea for this section.

Problem 3.1

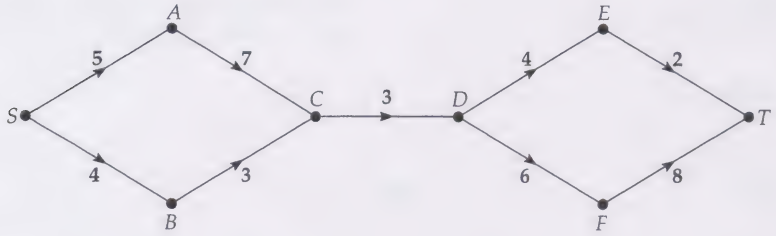
For each of the following networks, find the value of a maximum flow and the worst 'bottleneck'.

Note that a bottleneck may consist of more than one arc.

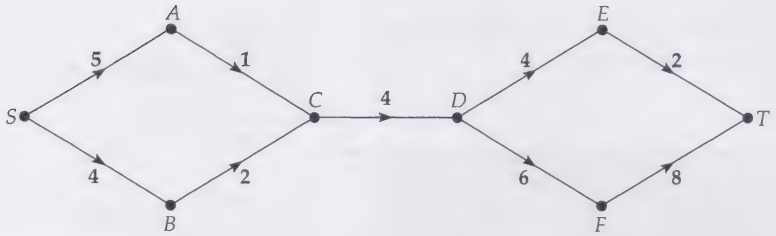
(a)



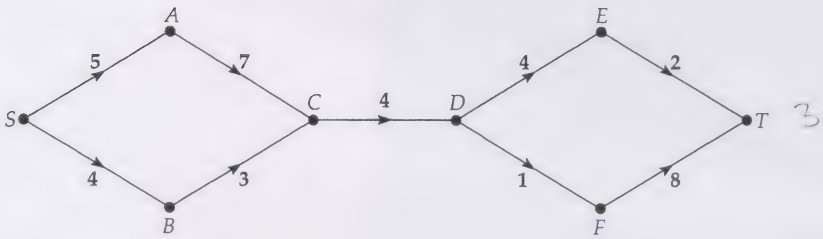
(b)



(c)



(d)



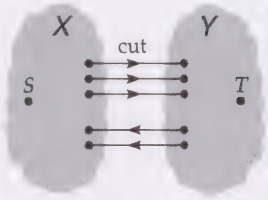
In each part of the solution to Problem 3.1, we considered a bottleneck consisting of a few arcs of small capacity through which the commodity in question has to flow. We can make this idea more precise by introducing the idea of a *cut*, and replacing the intuitive term 'worst bottleneck' by *minimum cut*.

Definitions

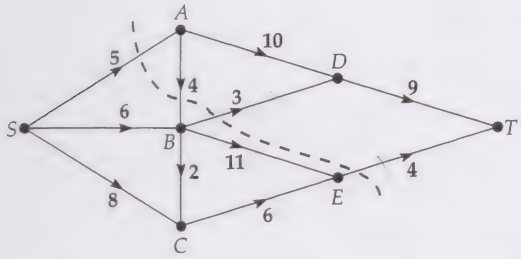
A **cut** in a basic network with source S and sink T is a set of arcs whose removal separates the network into two disjoint parts: X (containing S) and Y (containing T).

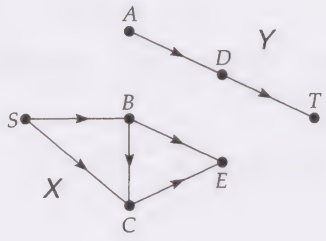
The **capacity of a cut** is the sum of the capacities of those arcs in the cut which are directed from X (the part containing S) to Y (the part containing T).

A **minimum cut** is a cut of smallest capacity.



Example 3.1





In this network, the arcs SA, AB, BD and ET form a cut (indicated by a broken line) whose removal separates the network into two disjoint parts, X (containing the vertices S, B, C and E) and Y (containing the vertices A, D and T).

Usually we specify the arcs in a cut, but sometimes we find it more convenient to give the two sets of vertices X and Y.

The capacitty of this cut is

$$5 + 3 + 4 = 12.$$

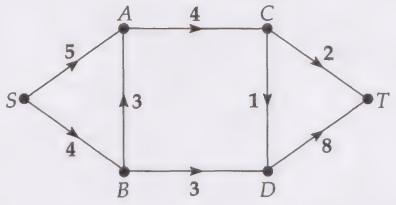
Another cut is given by the arcs SA, AB, BD, BE and CE, whose removal separates the network into two disjoint parts, X (containing the vertices S, B and C) and Y (containing the vertices A, D, E and T); this cut has capacity

$$5 + 3 + 11 + 6 = 25.$$

Note that the arc AB is *not included* in these calculations, since in each case it is directed *from* a vertex in Y *to* a vertex in X.

Every other cut in this network has capacity greater than 12, so the first cut $\{SA, AB, BD, ET\}$ is a minimum cut. ■

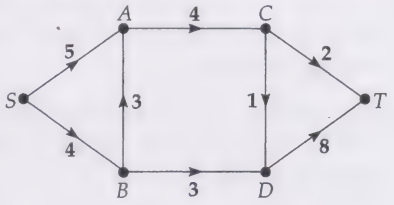
Example 3.2



For the above network, we can list all the cuts. In the following table we give the arcs in each cut, the vertices in X and Y, and the capacity of the cut.

	arcs in cut	X	Y	capacity of cut
1	SA, SB	S	A, B, C, D, T	$5+4=9$
2	SB, BA, AC	S, A	B, C, D, T	$4+4=8$
3	SA, BA, BD	S, B	A, C, D, T	$5+3+3=11$
4	AC, BD	S, A, B	C, D, T	$4+3=7$
5	SB, BA, CD, CT	S, A, C	B, D, T	$4+1+2=7$
6	SA, BA, CD, DT	S, B, D	A, C, T	$5+3+8=16$
7	BD, CD, CT	S, A, B, C	D, T	$3+1+2=6$
8	AC, CD, DT	S, A, B, D	C, T	$4+8=12$
9	CT, DT	S, A, B, C, D	T	$2+8=10$

Note that in each of the cuts (2), (5), (6) and (8), one of the arcs is directed from a vertex in Y to a vertex in X, and so we do not include it when calculating the capacity.

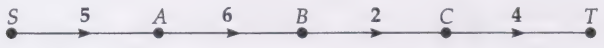


In this example there is only one minimum cut: $\{BD, CD, CT\}$, that is, cut (7) with capacity 6. ■

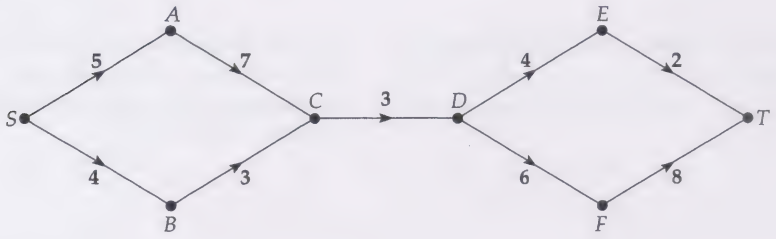
Problem 3.2

For each of the following networks, write down a minimum cut, its capacity, and the corresponding sets X and Y.

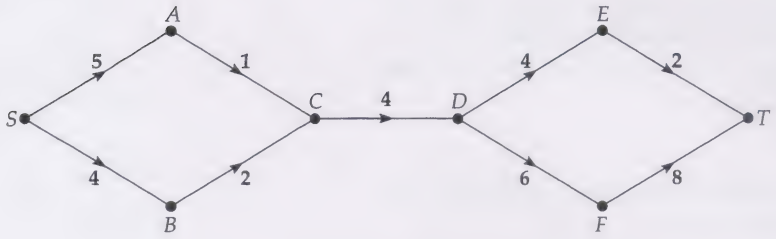
(a)



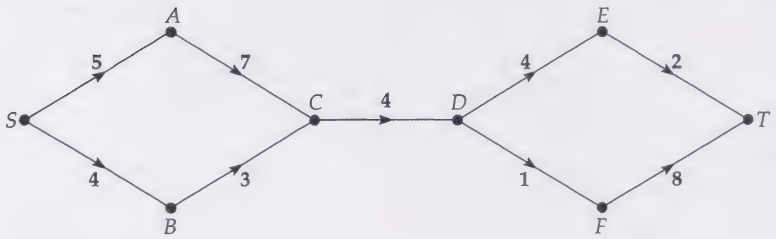
(b)



(c)

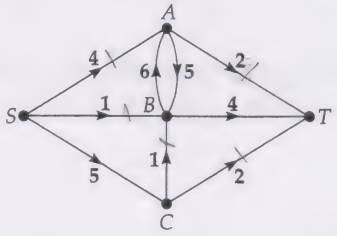


(d)



Problem 3.3

The network in the margin has eight cuts. Draw up a table (similar to that given in Example 3.2) listing these cuts, the vertices in X and Y, and the capacity of each cut. Which are the minimum cuts?



Problem 3.4

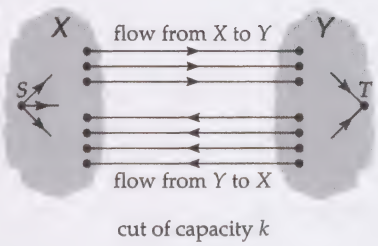
Construct an example of a network with two minimum cuts.

3.2 Max-flow min-cut theorem

Now that we have the concept of a *cut*, we can return to the problem of finding a maximum flow in a basic network. The connection between flows and cuts arises from the observation that

the value of any flow ≤ the capacity of any cut.

Why does this inequality always hold? Consider the diagram in the margin which illustrates a cut of capacity k.



cut of capacity k

The removal of the arcs in this cut separates the network into two disjoint parts, X and Y. But the value of a flow is the amount of the commodity flowing through the network, and all of this commodity must travel from X to Y along the arcs of the cut. So the value of the flow cannot exceed k, the maximum flow permitted through the cut.

If F denotes the value of the flow, then F is given by the equation

F = (the total flow from X to Y) – (the total flow from Y to X).

Since the total flow from X to Y is at most k, and the total flow from Y to X is at least 0, we have $F \leq k - 0$, that is, $F \leq k$.

However, there is nothing special about this flow and this cut, so we have

the value of *any* flow ≤ the capacity of *any* cut.

Now this inequality must be true for a *maximum* flow in particular. So

the value of any *maximum* flow ≤ the capacity of *any* cut.

But this is true for *any* cut, and so it must be true for a *minimum* cut in particular. So

the value of any *maximum* flow ≤ the capacity of any *minimum* cut.

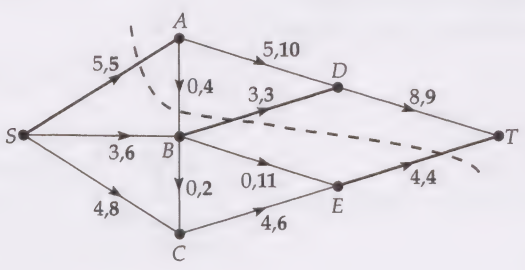
This last inequality is very important in practice.

Maximum flows and minimum cuts

If we can find a flow with value k, and a cut with capacity equal to this value of k, then:

- the flow is a maximum flow
  (since any larger flow would violate the inequality);
- the cut is a minimum cut
  (since any smaller cut would violate the inequality).

For example, the following diagram shows a network with a flow of value 12 and a cut of capacity 12.

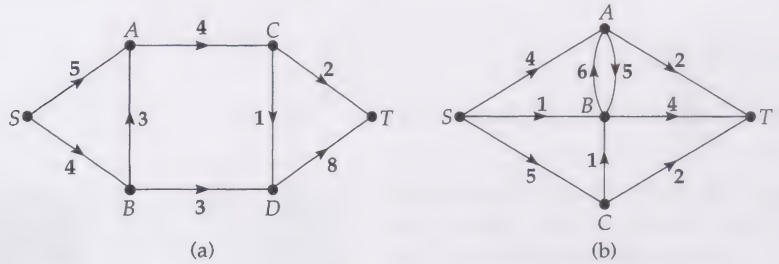


It follows that this flow is a maximum flow, and this cut is a minimum cut.

To see what can be deduced from the above inequalities, try the following problems.

Problem 3.5

For each of the following networks, find a flow and a cut for which the value of the flow is equal to the capacity of the cut. What can you deduce about the flow in each case?



(a) (b)

Hint Use the minimum cuts found in Example 3.2 and Problem 3.3.

Problem 3.6

(a) If you find a cut with capacity 7 in a network, what can you deduce about a maximum flow?

(b) If you find a flow with value 11 in a network, what can you deduce about a minimum cut?

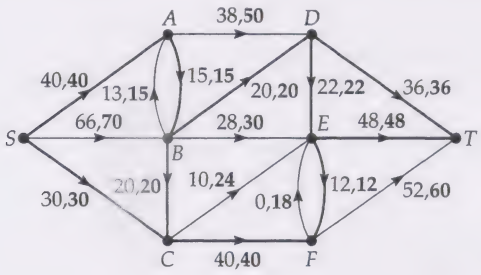
(c) If you find a flow with value 4 and a cut with capacity 6, what can you deduce?

(d) If you find a flow with value 6 and a cut with capacity 4, what can you deduce?

Problem 3.7

The following diagram illustrates a flow of value 136 in a network. Find a cut of capacity 136, and deduce that the given flow is a maximum flow.

This answers Problem 2.5 where we asked if a flow of value 124 is a maximum flow.



One observation you may have made about the solution to the above problem is that the value of a maximum flow *is equal to* the capacity of a minimum cut. This is not a coincidence, and the statement that these two numbers are always the same is known as the *max-flow min-cut theorem*. It is the central result in the study of network flows, and was first proved in this form by L. R. Ford and D. R. Fulkerson in 1956.

Theorem 3.1: max-flow min-cut theorem

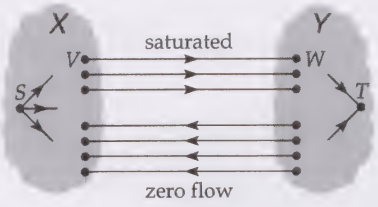
In any basic network, the value of a maximum flow is equal to the capacity of a minimum cut.

The max-flow min-cut theorem is an example of a 'minimax theorem' — the minimum of something = the maximum of something else. You will meet several minimax theorems in this course.

Proof

We have already shown that the value of any maximum flow can never exceed the capacity of any minimum cut. To show that these two numbers are actually equal, we need to find a cut whose capacity is equal to the value of any maximum flow.

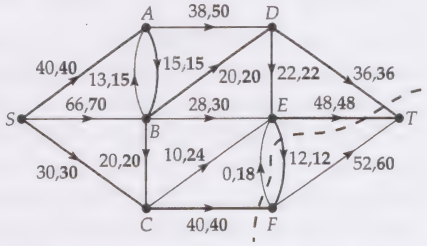
Consider any maximum flow. Let X be the set of all vertices which can be reached from S by a flow-augmenting path, and let Y be the set of all remaining vertices. Then T must lie in the set Y, because if T were in X then we could increase the flow from S to T, which is impossible since we started with a maximum flow.



We complete the proof by looking at the cut consisting of those arcs which have one end in X and the other end in Y. Any arc which is directed from a vertex V in X to a vertex W in Y must be saturated, since if it were not saturated there would be a flow-augmenting path from S to W, and W would have to be put into X instead of Y. By a similar argument, any arc which is directed from a vertex in Y to a vertex in X must carry a zero flow. So the capacity of the cut is equal to the total flow from X to Y, and this is equal to the value of the flow in the network. Since the flow is a maximum flow, this completes the proof. ■

An important consequence of this theorem is that, *if you can find a minimum cut in a network, then you can immediately deduce the value of a maximum flow.*

We illustrate the proof by the following example of a maximum flow.



The vertices A, B, C, D and E can all be reached from S by flow-augmenting paths, whereas F and T cannot. So $X = \{S, A, B, C, D, E\}$ and $Y = \{F, T\}$.

The cut in the above example consists of the saturated arcs CF, EF, DT and ET directed from X to Y, and the arc FE carrying a zero flow directed from Y to X.

The capacity of the above cut is $40 + 12 + 36 + 48 = 136$, which is the value of the above flow.

Problem 3.8

Decide whether each of the following statements is TRUE or FALSE, and give reasons for your answers.

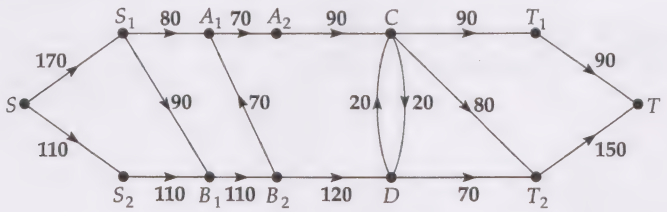
(a) Every minimum cut in a network carrying a maximum flow consists entirely of saturated arcs.

(b) If the capacity of every arc of a network is an integer, then the value of a maximum flow is also an integer.

Problem 3.9

Find a minimum cut in the following network.

Hint Use the maximum flow of value 160 found at the end of Section 2.3.



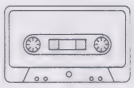
3.3 Maximum flow algorithm

In many examples, a minimum cut is easily found by inspection, and a maximum flow can then be found without much difficulty. However, for large and complicated networks, finding a minimum cut or maximum flow by inspection is impracticable, and a more systematic method is needed. In

this audio-tape section we describe a method for finding a maximum flow and a minimum cut in a basic network. The algorithm we discuss is called the *maximum flow algorithm*. It can be applied to a particular network either with pencil and paper (in the case of a small network) or by a computer (in the case of a large network).

The algorithm involves the alternate use of two parts: Part A, the *labelling procedure*, which labels appropriate vertices in order to identify flow-augmenting paths; and Part B, the *flow-augmenting procedure*, which increases the flow along flow-augmenting paths found in Part A. We say we have 'breakthrough' when the sink is labelled in Part A, and then proceed to Part B. The process stops when the labelling procedure indicates that no more flow-augmenting paths can be found: the resulting flow is then a maximum flow, and the labels at the final stage identify a minimum cut.

Now listen to band 1 of Audio-tape 1. You will need *Audio-tape Notes 1*.



3.4 Proof of Menger's theorem Not Assessed

We now prove Menger's theorem which we introduced in Section 1. We start by deducing the arc form for digraphs from the max-flow min-cut theorem for basic networks, and then show how the corresponding edge form for graphs follows immediately. Finally, we show how the vertex forms for both graphs and digraphs follow from these other versions.

We begin by restating the max-flow min-cut theorem.

Theorem 3.1: max-flow min-cut theorem

In any basic network, the value of a maximum flow is equal to the capacity of a minimum cut.

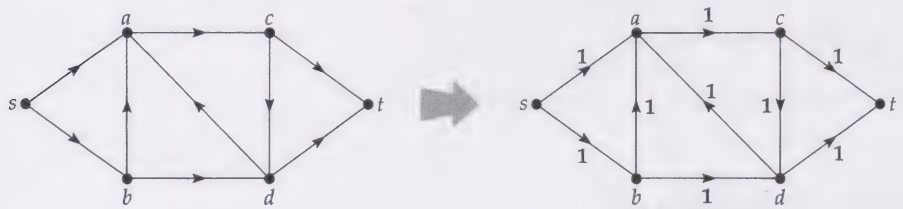
We deduce the arc form of Menger's theorem for digraphs.

Theorem 1.3: Menger's theorem for digraphs (arc form)

Let D be a connected digraph, and let s and t be vertices of D. Then the maximum number of arc-disjoint st-paths is equal to the minimum number of arcs separating s from t.

Proof

We associate with the digraph D a basic network N which is formed from D by giving each arc a capacity of 1, as indicated, for example, in the following diagram.



the digraph D the network N

Since each arc has capacity 1, a maximum flow from s to t in the network N must consist of a flow of value 1 along each of a set of arc-disjoint paths from s to t. Thus

the value of a *maximum* flow in N = the *maximum* number of arc-disjoint st-paths in D.

Now consider a set of arcs which correspond to a cut in the network N. Such a set of arcs separates s from t. Moreover, since the capacity of each arc of the network is 1, the capacity of the cut is the number of these arcs directed from the part containing s to that containing t. It follows that

For example, the set of arcs $\{ac, da, bd\}$ in the diagram above.

For example, two of the arcs ac, da and bd are so directed, and thus the capacity of the corresponding cut is 2.

the capacity of a *minimum* cut in N = the *minimum* number of arcs separating s from t in D.

But, by the max-flow min-cut theorem,

the value of a maximum flow in N = the capacity of a minimum cut in N,

and so

the maximum number of arc-disjoint st-paths in D = the minimum number of arcs separating s from t in D,

as required. ■

We can now deduce the edge form of Menger's theorem for graphs.

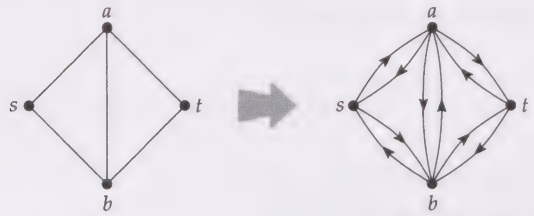
Theorem 1.2: Menger's theorem for graphs (edge form)

Let G be a connected graph, and let s and t be vertices of G. Then the maximum number of edge-disjoint st-paths is equal to the minimum number of edges separating s from t.

Outline of proof

We transform the graph G into a digraph $D(G)$ by replacing each edge by two arcs, one in each direction. This is indicated in the following diagram.

We used a similar transformation in Section 2.3.



the graph G the digraph $D(G)$

It can be shown that

the maximum number of edge-disjoint st-paths in G = the maximum number of arc-disjoint st-paths in $D(G)$,

and that

the minimum number of edges separating s from t in G = the minimum number of arcs separating s from t in $D(G)$.

But, by the arc form of Menger's theorem for digraphs (proved above),

the *maximum* number of arc-disjoint st-paths in $D(G)$ = the *minimum* number of arcs separating s from t in $D(G)$,

and so

the *maximum* number of edge-disjoint st-paths in G = the *minimum* number of edges separating s from t in G,

as required. ■

We can also deduce the vertex form of Menger's theorem for digraphs.

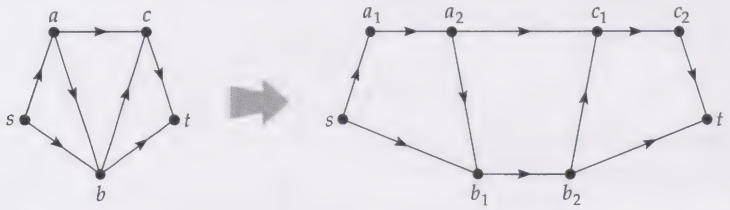
Theorem 1.5: Menger's theorem for digraphs (vertex form)

Let D be a connected digraph, and let s and t be non-adjacent vertices of D. Then the maximum number of vertex-disjoint st-paths is equal to the minimum number of vertices separating s from t.

Outline of proof

We transform the digraph D into another digraph D' by replacing each vertex v of D (other than s and t) by two vertices v_1 and v_2 joined by an arc. This is illustrated by the following diagram.

We used a similar transformation in Section 2.3.



the digraph D the digraph D'

All arcs of D directed towards a vertex v become arcs of D' directed towards the vertex v_1, and all arcs of D directed away from v become arcs of D' directed away from the vertex v_2.

It is not difficult to see that two or more st-paths in D are vertex-disjoint if and only if the corresponding st-paths in D' are arc-disjoint. Applying the arc form of Menger's theorem to D', we obtain the vertex form of Menger's theorem for D. ■

Finally, we can deduce the vertex form of Menger's theorem for graphs.

Theorem 1.4: Menger's theorem for graphs (vertex form)

Let G be a connected graph, and let s and t be non-adjacent vertices of G. Then the maximum number of vertex-disjoint st-paths is equal to the minimum number of vertices separating s from t.

Outline of proof

This form of Menger's theorem is deduced from the vertex form for digraphs in the same way as the edge form for graphs is deduced from the arc form for digraphs — namely, by consideration of the digraph $D(G)$. ■

3.5 Computer activities

The computer activities for this section are described in the *Computer Activities Booklet*.

After studying this section, you should be able to:

- explain the terms *cut*, *capacity of a cut* and *minimum cut*;
- explain why the value of any flow cannot exceed the capacity of any cut;
- state and use the max-flow min-cut theorem, and verify it for a given network;
- use the maximum flow algorithm to find maximum flows and minimum cuts;
- appreciate the connection between Menger's theorem and the max-flow min-cut theorem.

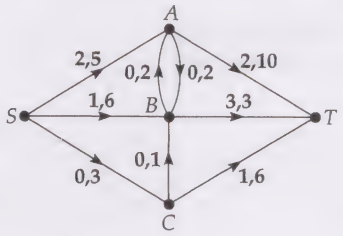
Section 4 Not Assessed

4 Networks with lower and upper capacities

In many practical problems (for example, the gas pipeline example mentioned earlier), it is necessary to specify both the maximum and the minimum allowable flow along each arc. We refer to the minimum allowable flow along an arc as the **lower capacity** of the arc, and the maximum allowable flow as the **upper capacity** of the arc.

A basic network can be regarded as a network of this type in which all lower capacities are zero.

The following diagram depicts a *network with lower and upper capacities* for each arc, the first number represents the lower capacity of the arc, and the second number represents the upper capacity.



In this section we discuss the problem of determining whether there is a *feasible* flow in such a network, and if so, how we can construct one.

4.1 Generalized flow problem

We begin by generalizing our definition of *flow*.

Definition

A **flow** in a network with lower and upper capacities is an assignment of non-negative numbers to the arcs in such a way that the following conditions are satisfied:

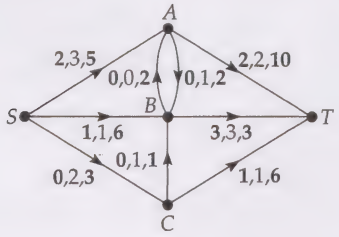
(a) the **feasibility condition**: the flow along each arc is not less than the lower capacity and is not more than the upper capacity of that arc, that is,

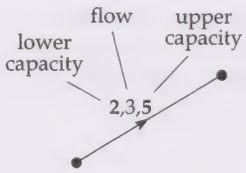
lower capacity ≤ flow ≤ upper capacity;

(b) the **flow conservation law**: the flow into each vertex (other than S or T) is equal to the flow out of it.

Terms such as the **value of a flow** and a **maximum flow** are defined in exactly the same way as before.

As with the basic flow problem, it is convenient to represent both flows and capacities on the same diagram. The flow appears between the lower capacity and the upper capacity, and the capacities are shown in bold.





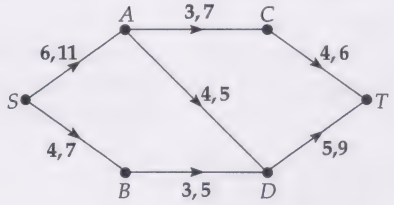
Problem 4.1

Check that the above flow satisfies the feasibility condition and the flow conservation law. What is the value of this flow?

The following worked problem illustrates how to find a maximum flow in a network with lower and upper capacities.

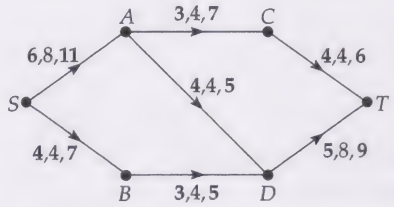
Worked problem

Find a maximum flow in the following network with lower and upper capacities.



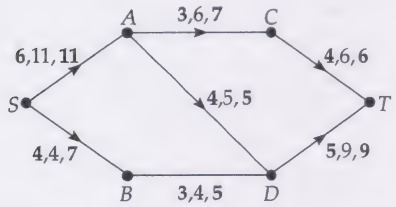
Solution

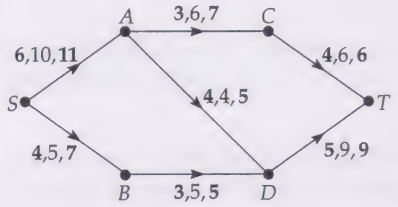
We begin by finding any possible flow. In this example we can do this by inspection. There are several possible flows in this network, for example, a flow of value 12:



Next, by considering flow-augmenting paths, we find a maximum flow in the network.

We can increase the flow by 2 along the path $SACT$, and by 1 along either of the paths $SADT$ and $SBDT$. This gives the two maximum flows of value 15 shown in the following diagrams.

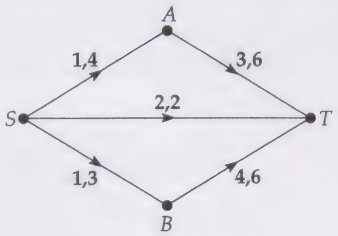




■

The above problem illustrates the fact that, if we can find a flow in a network with lower and upper capacities, then we can increase this flow to a maximum flow by means of the techniques described in Sections 2 and 3. In particular, we can use the maximum flow algorithm to increase the flow along flow-augmenting paths, provided that *when using a backward arc we do not decrease the flow in that arc to a value less than the lower capacity of the arc.*

However, there is a snag! Actually finding an initial flow is not always easy. (We can no longer choose the zero flow as the initial flow.) Indeed, there may not even be such a flow, as shown by the network in the margin.



It is impossible to find a flow in this network since the flow along SB cannot exceed 3, and the flow along BT must be at least 4. In such a network, the capacity restrictions are incompatible. This leads us to make the following definition.

Definition

A network with lower and upper capacities is **feasible** if there exists a flow satisfying all the capacity restrictions. If no such flow exists, then the network is **infeasible**.

In view of the above remarks, it would be useful to have a systematic method for testing whether a given network is feasible and, if it is, for finding a valid flow. In the second band of the audio-tape for this unit we present such a method, which we call the *feasibility algorithm*; it is useful

when we cannot find a flow by inspection. The method reduces the problem to that of finding a maximum flow in a related basic network — something we already know how to do.

Now listen to band 2 of Audio-tape 1.

Using the notation introduced in the audio-tape frames, we can now give a statement of the feasibility theorem, which forms the basis of the feasibility algorithm.

Theorem 4.1: feasibility theorem

A network N with lower and upper capacities is feasible if and only if in the related basic network N^* there is a flow from S^* to T^* such that all the arcs out of S^* and all the arcs into T^* are saturated.

4.2 Generalized max-flow min-cut theorem

We conclude this section by stating an analogue of the max-flow min-cut theorem for networks with lower and upper capacities. First, we need to generalize our definition of the capacity of a cut.

Definition

Let N be a network with lower and upper capacities, and let C be a cut which separates N into two disjoint parts: X (containing S) and Y (containing T). Then the **capacity of the cut C** is

(the sum of the upper capacities of the arcs of C which are directed from X to Y)
– (the sum of the lower capacities of the arcs of C which are directed from Y to X).

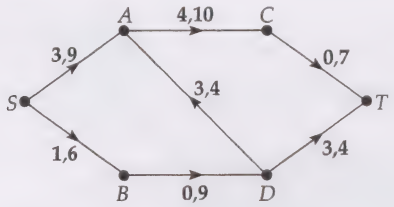
Note that if the lower capacity of each arc is zero, then this definition coincides with the definition given in Section 3.1.

For example, for the network N shown below,

the capacity of the cut $\{SA, SB\}$ is $9 + 6 = 15$;

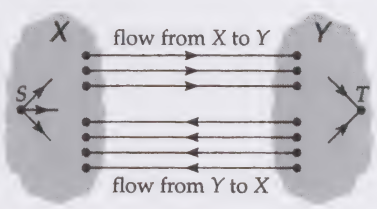
the capacity of the cut $\{AC, DA, BD\}$ is $(10 + 9) - 3 = 16$;

the capacity of the cut $\{SB, DA, CT\}$ is $(6 + 7) - 3 = 10$.



By an analogous method to that given in Section 3.2, we can show that, in any network with lower and upper capacities, the value of any flow from S to T does not exceed the capacity of any cut.

Consider a cut C with capacity k, say. The removal of the arcs of C separates the network into two parts, X and Y. If the sum of the upper capacities of the arcs of C directed from X to Y is k_u, and the sum of the lower capacities of the arcs of C directed from Y to X is k_l, then $k_u - k_l = k$.



cut of capacity k

If F denotes the value of the flow, then F is given by the equation

F = (the total flow from X to Y) – (the total flow from Y to X).

Since the total flow from X to Y is at most k_u, and the total flow from Y to X is at least k_l, we have $F \leq k_u - k_l$, that is, $F \leq k$.

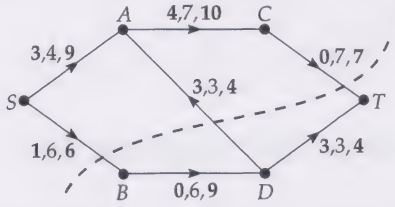
Just as before, it is possible to show that the value of any maximum flow from S to T is equal to the capacity of any minimum cut.

Theorem 4.2: generalized max-flow min-cut theorem

In any feasible network with lower and upper capacities, the value of a maximum flow is equal to the capacity of a minimum cut.

The proof of this theorem is very similar to that of the max-flow min-cut theorem, and we omit it.

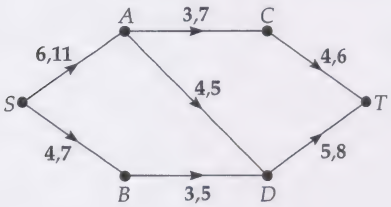
We can use this theorem to find a *minimum cut* and a *maximum flow* in a particular network by finding a flow and a cut with the property that *the value of the flow is equal to the capacity of the cut*. Then the cut is a minimum cut and the flow is a maximum flow. For example, we can construct a flow of value 10 in the above network:



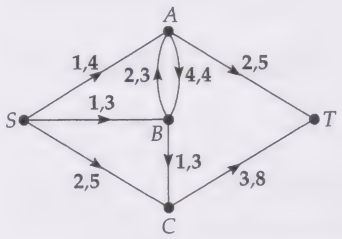
Since $\{SB, DA, CT\}$ is a cut of capacity $(6 + 7) - 3 = 10$, it follows that this cut is a minimum cut, and that the above flow is a maximum flow of value $4 + 6 = 7 + 3 = 10$.

Problem 4.2

Verify the generalized max-flow min-cut theorem for each of the following feasible networks; the numbers next to an arc are the lower and upper capacities.



(a)



(b)

4.3 Computer activities

The computer activities for this section are described in the *Computer Activities Booklet*.

After studying this section, you should be able to:

- explain the terms *network with lower and upper capacities, feasible network* and *infeasible network;*
- use the feasibility algorithm to determine whether a given network with lower and upper capacities is feasible and, if it is, find a maximum flow;
- find the capacity of a cut and a minimum cut in a network with lower and upper capacities;
- state and use the generalized max-flow min-cut theorem for feasible networks.

Further reading

The material covered in this unit is included in a number of books, for example, the following.

J. Clark and D. A. Holton, *A First Look at Graph Theory*, World Scientific Pub. Co., 1991.

G. Chartrand and O. R. Oellermann, *Applied and Algorithmic Graph Theory*, McGraw-Hill, 1993.

B. Carré, *Graphs and Networks*, Clarendon Press, 1979.

N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, 1974.

N. Christofides, *Graph Theory: An Algorithmic Approach*, Academic Press, 1975.

C. L. Liu, *Introduction to Combinatorial Mathematics*, McGraw-Hill, 1968.

Some uses of network theory are described in the following.

V. Chachra, P. M. Ghare and J. M. Moore, *Applications of Graph Theory Algorithms*, Elsevier/North Holland, 1979.

Finally, a more advanced treatment is found in the classic book of Ford and Fulkerson:

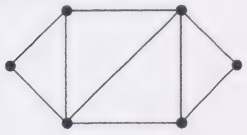
L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

Exercises

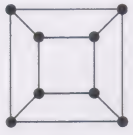
Section 1

Connectivity of graphs

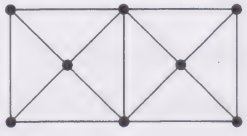
1.1 Write down the values of $\kappa(G)$ and $\lambda(G)$ for each of the following graphs G.



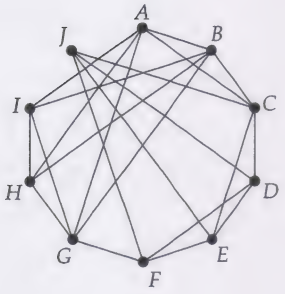
(a)



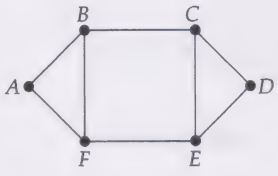
(b)



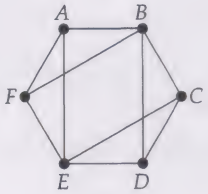
(c)



(d)



(e)



(f)

1.2 Give an example (where possible) of a graph G for which:

(a) $\kappa(G) = 2$, $\lambda(G) = 3$, $\delta(G) = 4$;

(b) $\kappa(G) = 3$, $\lambda(G) = 2$, $\delta(G) = 4$;

(c) $\kappa(G) = 2$, $\lambda(G) = 2$, $\delta(G) = 4$.

1.3 In the Petersen graph, find:

(a) a cutset with 3 edges;

(b) a cutset with 4 edges;

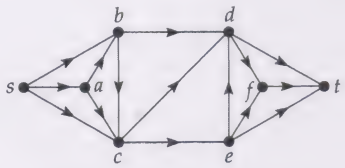
(c) a cutset with 5 edges;

(d) a cutset with 6 edges.

Menger's theorem and its analogues

1.4

(a) By finding k arc-disjoint st-paths and k arcs separating s from t (for the same value of k), find the maximum number of arc-disjoint st-paths in the following digraph.



(b) Similarly, find the maximum number of vertex-disjoint st-paths in the above digraph.

1.5 In the complete bipartite graph $K_{10,13}$, let v be any vertex in the set with 10 vertices, and w be any vertex in the other set.

(a) Find the maximum number of edge-disjoint paths between v and w.

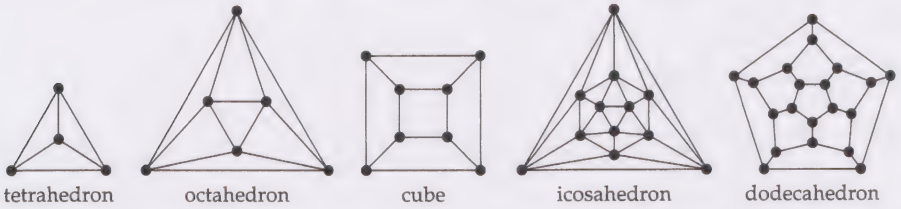
(b) Find the maximum number of vertex-disjoint paths between v and w.

Hint First remove the edge vw, apply the vertex form of Menger's theorem, then restore the edge vw.

Optimal connectivity

1.6 Which of the graphs in Exercise 1.1 have optimal connectivity?

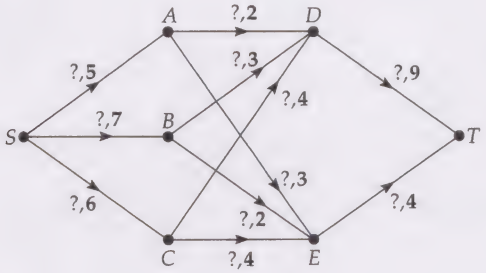
1.7 Which of the Platonic graphs have optimal connectivity?



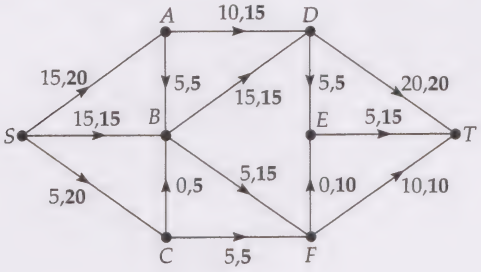
Section 2

Flows in basic networks

2.1 In the following basic network, the arcs AD, BE, CD and ET are saturated, and the flows along SA, SB and SC are all the same. Find all the missing flows.



2.2 Consider the following flow in a basic network.



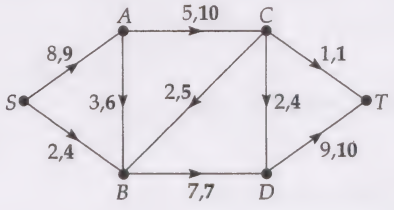
(a) Which of the arcs are saturated?

(b) What is the value of the flow from S to T?

(c) There is just one flow-augmenting path involving only forward arcs; find it, and determine the amount by which the flow along it can be increased. What is the value of the new flow?

(d) With this new flow, there is just one flow-augmenting path (involving backward arcs); find it, and hence determine the value of a maximum flow in the network.

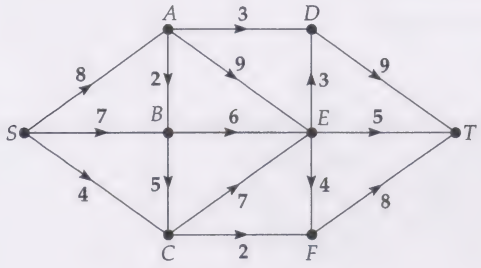
2.3 Consider the following flow of value 10 in a basic network.



(a) Find all the flow-augmenting paths.

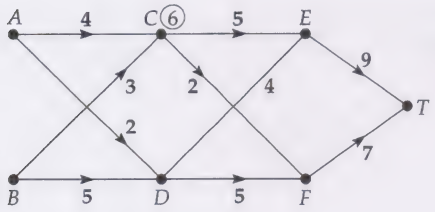
(b) What is the value of a maximum flow?

2.4 By identifying flow-augmenting paths, find a maximum flow in the following basic network.

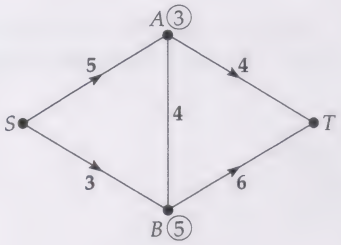


Transforming to a basic network

2.5 Transform each of the following networks into a basic network.

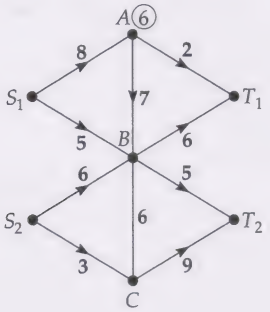


(a)



(b)

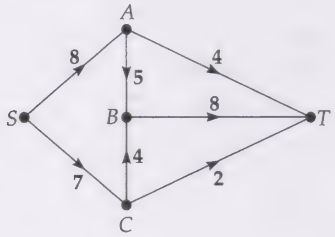
2.6 Transform the following network into a basic network.



Section 3

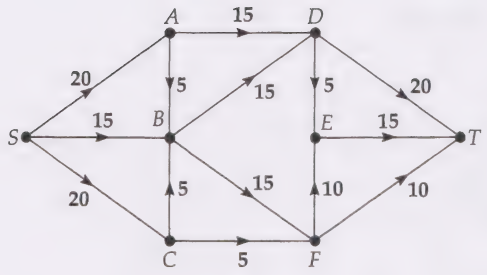
Max-flow min-cut theorem

3.1 Consider the following basic network.

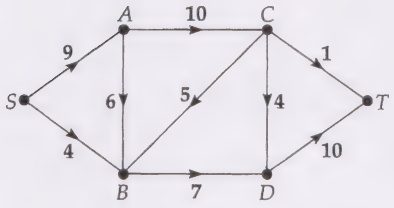


Draw up a table listing all the cuts, the vertices in the sets X and Y, and the capacity of each cut. Which are the minimum cuts?

3.2 For each of the following networks, find as many cuts as you can where, for each cut, the capacity is equal to the value of a maximum flow. (You found a maximum flow for (a) in Exercise 2.2 and for (b) in Exercise 2.3.) Are these cuts necessarily minimum cuts?



(a)



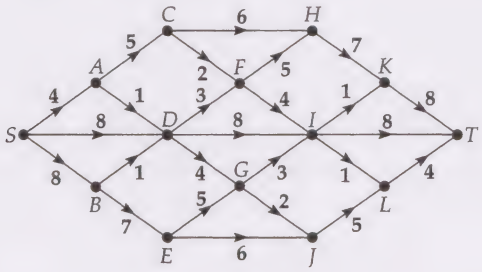
(b)

3.3 Classify each of the following statements as TRUE or FALSE.

(a) If a cut in a basic network consists entirely of saturated arcs for some flow in the network, then it must be a minimum cut.

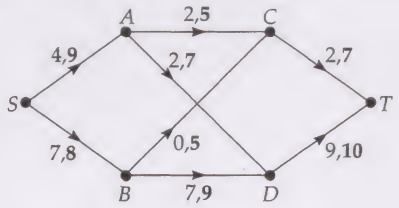
(b) If all the arcs in a network have different capacities, then there is only one minimum cut.

3.4 Find a flow of value 20 in the following basic network. Is this a maximum flow?

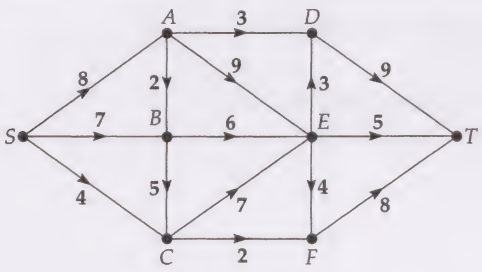


Maximum flow algorithm

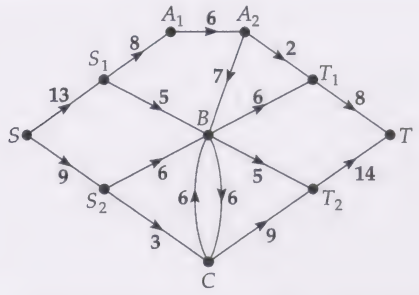
3.5 Use the algorithm to find a maximum flow and a minimum cut in the following basic network, both directly and by using the tabular method.



3.6 Use the algorithm to find a maximum flow and a minimum cut in the following basic network, where the numbers indicate capacities.



3.7 Find a maximum flow in the following basic network.

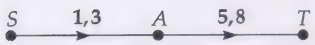


Hence find a maximum flow in the network of Exercise 2.6.

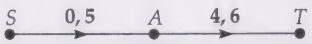
Section 4

Networks with lower and upper capacities

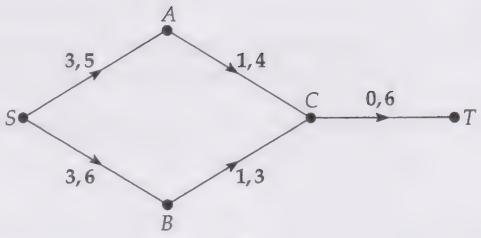
4.1 Determine by inspection which of the following networks are feasible.



(a)

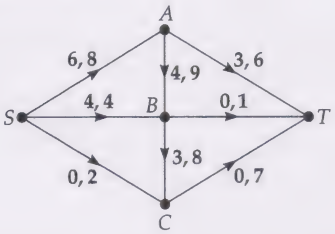


(b)



(c)

4.2 Consider the following network.

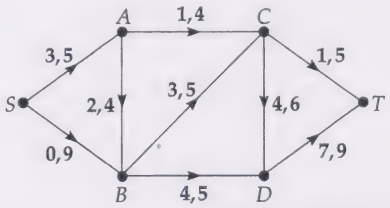


(a) By inspection, determine a feasible flow in this network.

(b) By increasing the flow systematically, find a maximum flow in this network.

(c) Find a minimum cut in this network.

4.3 Repeat Exercise 4.2 for the following network.



4.4 Use the feasibility theorem to check your answers to Exercise 4.1.

4.5 Use the feasibility theorem to show that the network in Exercise 4.2 is feasible, and hence find a feasible flow.

4.6 Use the feasibility theorem to show that the network in Exercise 4.3 is feasible, and hence find a feasible flow.

Solutions to the exercises

1.1

(a) $\kappa(G) = 2, \lambda(G) = 2$; (b) $\kappa(G) = 3, \lambda(G) = 3$;

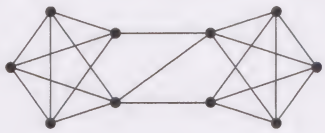
(c) $\kappa(G) = 2, \lambda(G) = 3$; (d) $\kappa(G) = 2, \lambda(G) = 3$;

(e) $\kappa(G) = 2, \lambda(G) = 2$; (f) $\kappa(G) = 2, \lambda(G) = 3$.

The graph in part (d) can be disconnected by removing, for example, the two vertices C and G or the three edges AC, BC, FG.

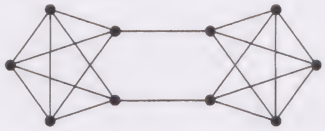
1.2

(a) There are several possibilities, for example:

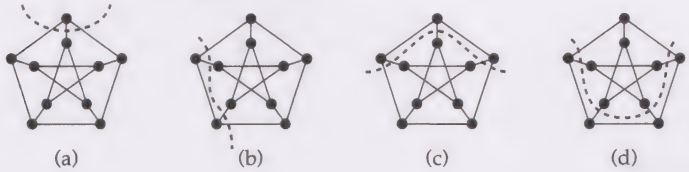


(b) No such graph exists since $\lambda(G) < \kappa(G)$, contradicting Theorem 1.1.

(c) There are several possibilities, for example:

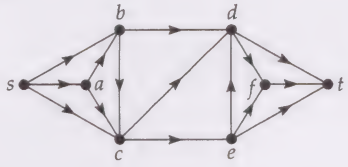


1.3 There are several possibilities, for example:



1.4

(a) Three arc-disjoint st-paths are $sbdt$, $scet$ and $sacdft$, and three arcs separating s from t are sa, sb and sc. Thus the maximum number of arc-disjoint st-paths and the minimum number of arcs separating s from t are both equal to 3.

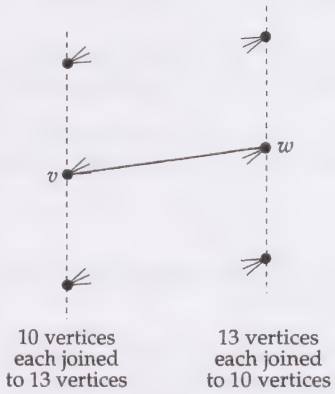


(b) Two vertex-disjoint st-paths are $sbdt$ and $scet$, and two vertices separating s from t are c and d. Thus the maximum number of vertex-disjoint st-paths and the minimum number of vertices separating s from t are both equal to 2.

1.5

(a) By Menger's theorem for graphs (edge form), the maximum number of edge-disjoint paths between v and w is equal to the minimum number of edges separating v from w, namely 10, since there are 10 edges incident with w.

(b) First we remove the edge vw, so that we can apply Menger's theorem to *non-adjacent* vertices. By Menger's theorem for graphs (vertex form), the maximum number of vertex-disjoint paths between v and w in the modified graph is equal to the minimum number of vertices separating v from w, namely 9. Now we restore the edge vw, which gives one more path. Thus the maximum number of vertex-disjoint paths between v and w is 10.



1.6 Only graph (b) has optimal connectivity:

for graph (a), $\kappa(G) = 2$, but $2m/n = 18/6 \neq 2$;

for graph (b), $\kappa(G) = 2m/n = 24/8 = 3$;

for graph (c), $\kappa(G) \neq \lambda(G)$;

for graph (d), $\kappa(G) \neq \lambda(G)$;

for graph (e), $\kappa(G) = 2$, but $2m/n = 16/6 \neq 2$;

for graph (f), $\kappa(G) \neq \lambda(G)$.

Alternatively, graphs (a), (c), (d), (e) and (f) cannot have optimal connectivety, because they are not regular.

1.7 All the Platonic graphs have optimal connectivity:

for the tetrahedron, $\kappa(G) = 2m/n = 3$;

for the octahedron, $\kappa(G) = 2m/n = 4$;

for the cube, $\kappa(G) = 2m/n = 3$;

for the icosahedron, $\kappa(G) = 2m/n = 5$;

for the dodecahedron, $\kappa(G) = 2m/n = 3$.

2.1 Since the arcs AD, BE, CD and ET are saturated, we have:

flow along $AD = 2$;

flow along $BE = 2$;

flow along $CD = 4$;

flow along $ET = 4$.

If the flow along each of SA, SB and SC is x, then applying the flow conservation law at A, B, C and D, we have:

flow along $AE = x - 2$;

flow along $BD = x - 2$;

flow along $CE = x - 4$;

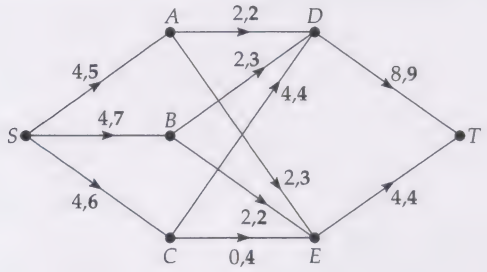
flow along $DT = 2 + (x - 2) + 4 = x + 4$.

Applying the flow conservation law at E, we obtain

$$(x - 2) + 2 + (x - 4) = 4,$$

so $x = 4$.

The flows are therefore as shown.



2.2

(a) The saturated arcs are SB, AB, BD, CF, DE, DT and FT.

(b) The value of the flow is $15 + 15 + 5 = 20 + 5 + 10 = 35$.

(c) The flow-augmenting path is $SCBFET$, with a maximum increase in flow of 5; the resulting flow has value 40.

(d) The flow-augmenting path is $SADBFET$, with a maximum increase in flow of 5; the value of a maximum flow is therefore 45.

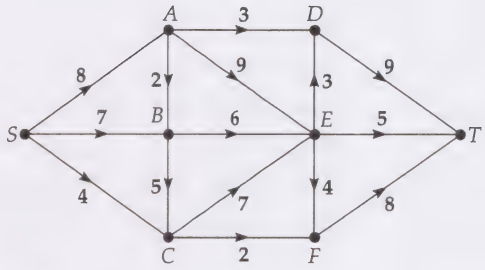
2.3

(a) The flow-augmenting paths are $SACDT$, $SBCDT$, $SABCDT$ and $SBACDT$.

(b) Each of these flow-augmenting paths includes the arc DT, which limits the increase in flow to 1. The value of a maximum flow is therefore $10 + 1 = 11$.

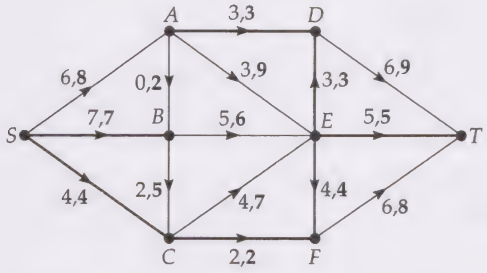
2.4 To find a maximum flow, consider the following flow-augmenting paths.

flow-augmenting path	increase in flow
SADT	3
SBET	5
SCFT	2
SAEDT	3
SBCEFT	2
SCEFT	2



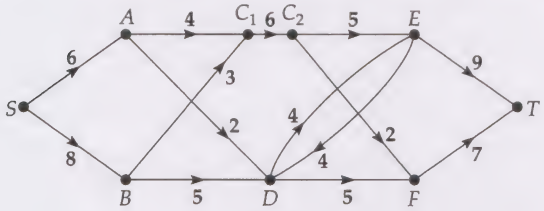
Other choices of flow-augmenting paths are possible.

This gives the following flow with value 17.

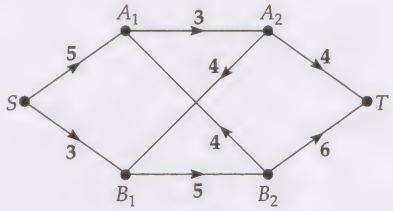


Since the arcs *AD*, *ED*, *ET*, *EF* and *CF* are all saturated, no further increase in the value of the flow is possible, so the above flow is a maximum flow.

2.5 The basic networks corresponding to the given networks are:



(a)

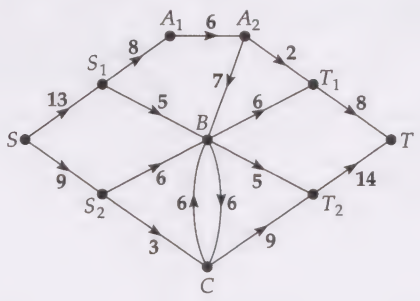


(b)

For network (a), we replace the undirected edge *DE* by two arcs of the same capacity, and replace the vertex *C* with an arc of the same capacity. We then add a source *S*, an arc *SA* of capacity 6 (the sum of the capacities of the arcs out of *A*) and an arc *SB* of capacity 8 (the sum of the capacities of the arcs out of *B*).

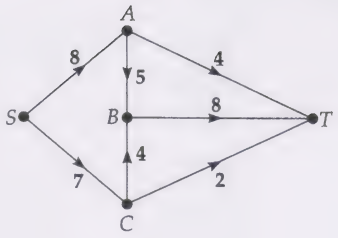
For network (b), we replace the undirected edge *AB* by two arcs of the same capacity: we replace the vertex *A* by an arc A_1A_2 of the same capacity, and we replace the vertex *B* by an arc B_1B_2 of the same capacity. The flow originally into vertex *A* goes in at A_1, but the flow originally out of *A* now comes out at A_2; and similarly for vertex *B*. Thus the two new arcs must be B_2A_1 and A_2B_1, as shown above.

2.6 In this case, we replace the undirected edge *BC* by two arcs, and replace the vertex *A* by an arc A_1A_2. We then add a source *S* and a sink *T*, together with arcs SS_1, SS_2, T_1T and T_2T.



3.1

	arcs in cut	X	Y	capacity of cut
1	SA, SC	S	A, B, C, T	$8+7=15$
2	SC, AB, AT	S, A	B, C, T	$7+5+4=16$
3	SA, AB, BT, CB, SC	S, B	A, C, T	$8+8+7=23$
4	SA, CB, CT	S, C	A, B, T	$8+4+2=14$
5	SC, CB, BT, AT	S, A, B	C, T	$7+4+8=19$
6	AT, AB, CB, CT	S, A, C	B, T	$4+5+4+2=15$
7	SA, AB, BT, CT	S, B, C	A, T	$8+8+2=18$
8	AT, BT, CT	S, A, B, C	T	$4+8+2=14$

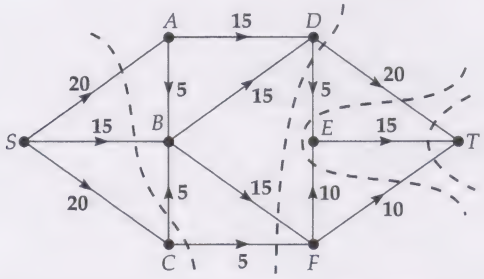


The minimum cuts are those with capacity 14: cuts (4) and (8).

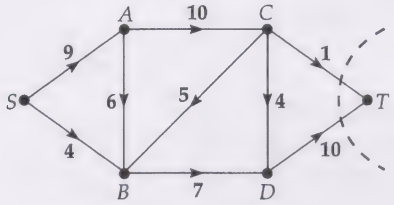
3.2

(a) There are five cuts with capacity 45:

$\{SA, SB, CB, CF\}$, $\{AD, AB, SB, CB, CF\}$, $\{DT, DE, BF, CF\}$, $\{DT, DE, FE, FT\}$, $\{DT, ET, FT\}$. All but the second of these are shown below.



(b) There is only one cut with capacity 11: $\{CT, DT\}$.

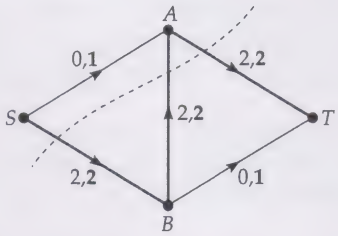


All of these cuts are minimum cuts, by the remarks preceding Problem 3.5 (or equivalently by the max-flow min-cut theorem).

3.3

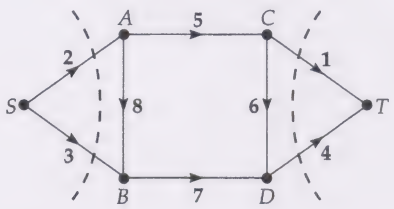
(a) The statement is FALSE. For example, consider the network shown in the margin.

The cut $\{SB, BA, AT\}$ of capacity 4 consists entirely of saturated arcs, but is not a minimum cut since each of the cuts $\{SA, SB\}$ and $\{AT, BT\}$ has capacity 3. (Note that this flow is not a maximum flow, since $SABT$ is a flow-augmenting path.)



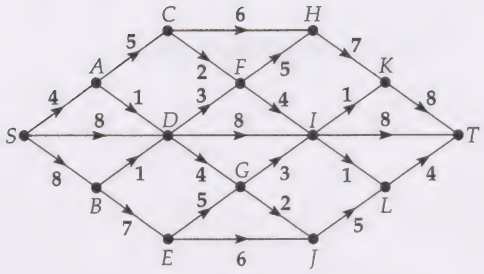
The style of the above cut is different as it is not a minimum cut.

(b) The statement is FALSE. For example, consider the following network.



There are two minimum cuts with capacity 5: $\{SA, SB\}$ and $\{CT, CD\}$.

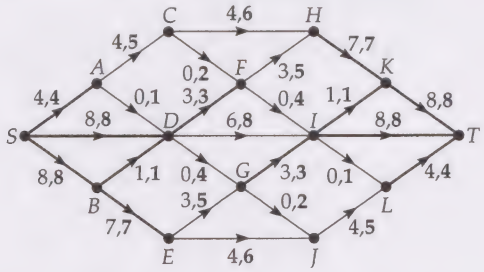
3.4



To find a maximum flow, consider the following flow-augmenting paths.

flow-augmenting path	flow increase
SACHKT	4
SDIT	8
SBEJLT	4
SBDFHKT	1
SBEGIKT	1
SBEGIDFHKT	2

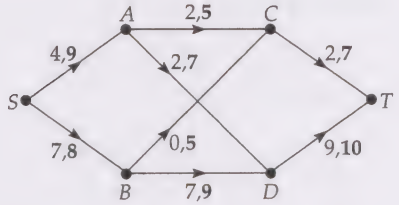
This gives the following flow with value 20.



Other choices of flow-augmenting paths are possible.

There exists a cut with capacity 20, for example {*SA*, *SD*, *SB*}; so, by the max-flow min-cut theorem, the above flow is a maximum flow.

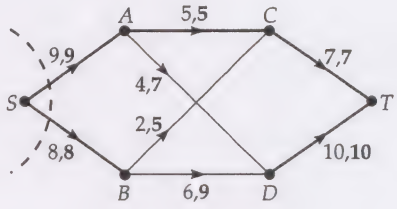
3.5



There is already a flow of value 11. We find flow-augmenting paths as follows.

iteration	labels	path	flow increase
1	*A*(*S*,5), *C*(*A*,3), *T*(*C*,3)	*SACT*	3
2	*A*(*S*,2), *D*(*A*,2), *T*(*D*,1)	*SADT*	1
3	*A*(*S*,1), *D*(*A*,1), *B*(*D*−,1), *C*(*B*,1), *T*(*C*,1)	*SADBCT*	1
4	*B*(*S*,1), *C*(*B*,1), *T*(*C*,1)	*SBCT*	1
5	none		no breakthrough

Thus we find a maximum flow of value 17. At the fifth iteration, when there is no breakthrough, no labelling is possible. It follows that there is a minimum cut of capacity 17 which separates *S* from the other vertices, namely {*SA*, *SB*}.



Alternatively, using the tabular method, we obtain the following.

Part A: label vertices **Part B: augment flow**

1

	A	*B*	*C*	*D*	*T*
S	4,**9**	7,**8**			
A			2,**5**	2,**7**	
B			0,**5**	7,**9**	
C					2,**7**
D					9,**10**
	(*S*,5)		(*A*,3)		(*C*,3)

Increase flow along *SACT* by 3.

2

	A	*B*	*C*	*D*	*T*
S	7,**9**	7,**8**			
A			5,**5**	2,**7**	
B			0,**5**	7,**9**	
C					5,**7**
D					9,**10**
	(*S*,2)			(*A*,2)	(*D*,1)

Increase flow along *SADT* by 1.

3

	A	*B*	*C*	*D*	*T*	
S	8,**9**	7,**8**				
A			5,**5**	3,**7**		
B			0,**5**	7,**9**		(*D*–,1)
C					5,**7**	
D					10,**10**	
	(*S*,1)		(*B*,1)	(*A*,1)	(*C*,1)	

Increase flow along *SADBCT* by 1.

4

	A	*B*	*C*	*D*	*T*
S	9,**9**	7,**8**			
A			5,**5**	4,**7**	
B			1,**5**	6,**9**	
C					6,**7**
D					10,**10**
		(*S*,1)	(*B*,1)		(*C*,1)

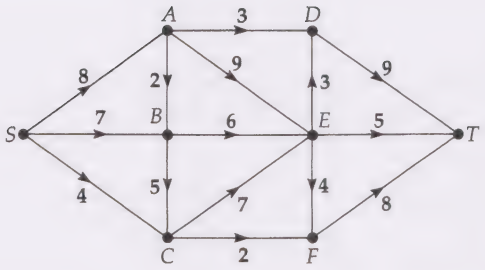
Increase flow along *SBCT* by 1.

5

	A	*B*	*C*	*D*	*T*
S	9,**9**	8,**8**			
A			5,**5**	4,**7**	
B			2,**5**	6,**9**	
C					7,**7**
D					10,**10**

No more flow-augmenting paths.
Maximum flow: 7 + 10 = 17.
Minimum cut {*S*}, {*A*, *B*, *C*, *D*, *T*} has capacity 9 + 8 = 17.

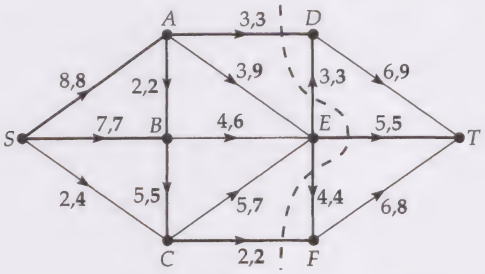
3.6



We find flow-augmenting paths as follows.

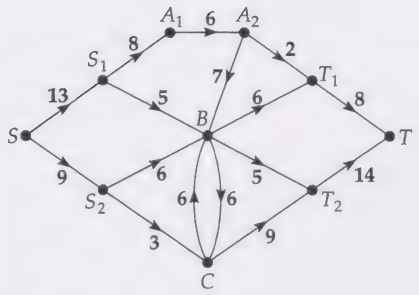
iteration	labels	path	flow increase
1	$A(S,8)$, $B(A,2)$, $C(B,2)$, $E(C,2)$, $D(E,2)$, $T(D,2)$	$SABCEDT$	2
2	$A(S,6)$, $D(A,3)$, $T(D,3)$	$SADT$	3
3	$A(S,3)$, $E(A,3)$, $D(E,1)$, $T(D,1)$	$SAEDT$	1
4	$A(S,2)$, $E(A,2)$, $F(E,2)$, $T(F,2)$	$SAEFT$	2
5	$B(S,7)$, $C(B,3)$, $E(C,3)$, $F(E,2)$, $T(F,2)$	$SBCEFT$	2
6	$B(S,5)$, $C(B,1)$, $E(C,1)$, $T(E,1)$	$SBCET$	1
7	$B(S,4)$, $E(B,4)$, $T(E,4)$	$SBET$	4
8	$C(S,4)$, ~~$E(C,2)$~~, ~~$A(E-,2)$~~, ~~$B(E-,2)$~~, $F(C,2)$, $T(F,2)$	$SCFT$	2
9	$C(S,2)$, ~~$E(C,2)$~~, ~~$A(E-,2)$~~, ~~$B(E-,2)$~~		no breakthrough

Thus we find a maximum flow of value 17. At the ninth iteration, when there is no breakthrough, vertices A, B, C and E are labelled; it follows that there is a minimum cut which separates S, A, B, C and E from the other vertices, namely $\{AD, ED, ET, EF, CF\}$ with capacity $3+3+5+4+2=17$.



Alternatively, using the tabular method, we would get the same labels and flow-augmenting paths.

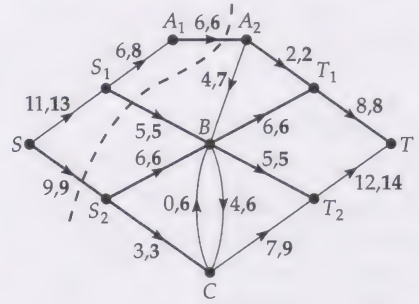
3.7



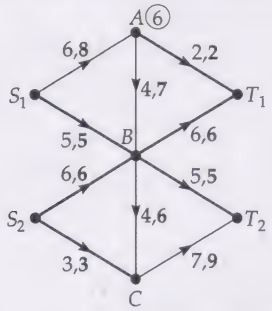
To find a maximum flow, consider the following flow-augmenting paths.

flow-augmenting path	flow increase
$SS_1A_1A_2T_1T$	2
$SS_1A_1A_2BT_1T$	4
SS_1BT_1T	2
SS_1BT_2T	3
SS_2BT_2T	2
SS_2BCT_2T	4
SS_2CT_2T	3

This gives the following flow with value 20. This is a maximum flow, since there exists a cut with capacity 20 — namely, $\{SS_2, S_1B, A_1A_2\}$.



A maximum flow in the original network of Exercise 2.6 is as follows.



An alternative solution is possible with BT_1 carrying a flow of 4, BC carrying a flow of 6, and CT_2 carrying a flow of 9.

4.1

(a) Infeasible: SA cannot take a flow of value greater than 3, and AT cannot take a flow of value less than 5.

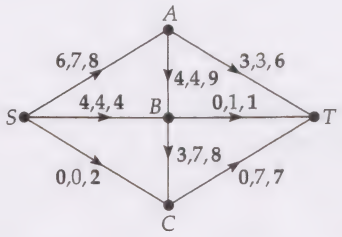
(b) Feasible — for example, a flow of 4 along both arcs;

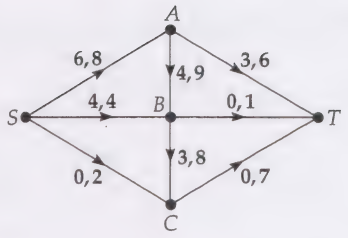
(c) Feasible — for example, a flow of 6 along the arc CT, and a flow of 3 along all the other arcs.

4.2

(a) The arc AB must carry a flow of at least 4; the arc AT must carry a flow of at least 3, so the arc SA must carry a flow of 7 or 8.

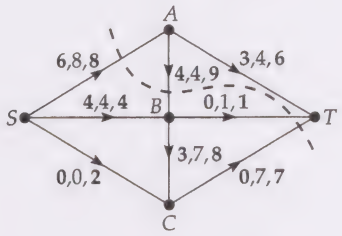
Let us try a flow of value 7 along SA; this gives the following feasible flow with value 11.





(b) We can increase the value of the flow along the path SAT by 1. No further increase in flow is possible. Thus a maximum flow of value 12 is as follows.

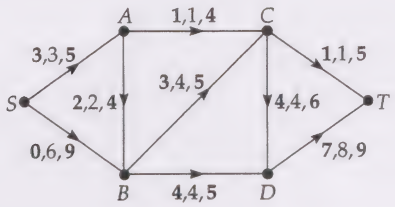
We cannot use BA as a backward arc, because it must carry a flow of value at least 4.

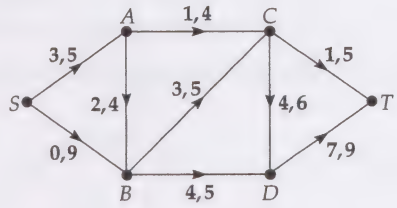


(c) A minimum cut is $\{SA, AB, BT, CT\}$ with capacity $(8 + 1 + 7) - 4 = 12$.

4.3

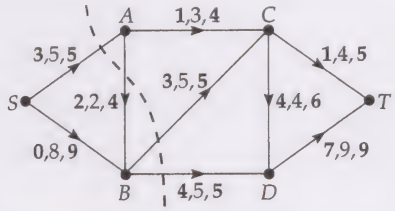
(a) There are many possibilities; for example, the following diagram shows a feasible flow of value 9.





(b) We can increase the value of the flow along $SACT$ by 2, along $SBCT$ by 1, and along $SBDT$ by 1. Or, alternatively, along $SACT$ by 2, along $SBCDT$ by 1, and along $SBDCT$ by 1.

Thus a maximum flow of value 13 is as follows.

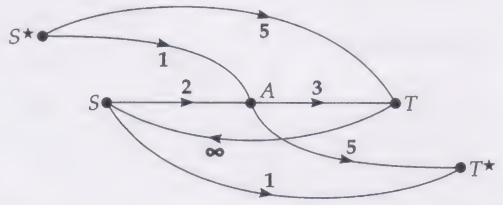


(c) A minimum cut is ~~$\{CT, DT\}$~~ with capacity $(5 + 5 + 5) - 2 = 13$.

$\{SA, AB, BC, BD\}$

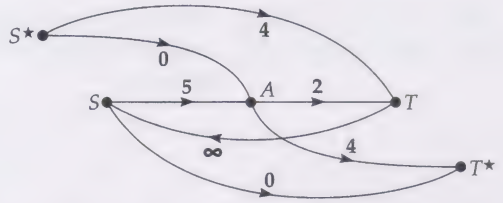
4.4

(a) The corresponding basic network N^* is as follows.

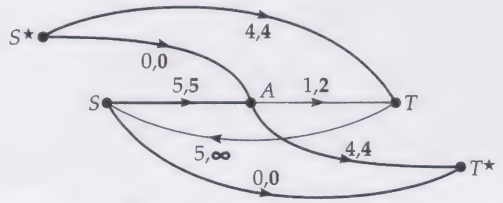


The maximum flow into A is 3, so there is no possible flow from S^* to T^* in which the arc AT^* is saturated. Thus, by the feasibility theorem, the original network is infeasible.

(b) The corresponding basic network N^* is as follows.

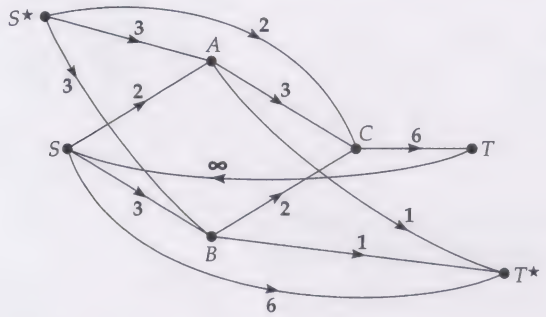


A suitable flow from S^* to T^* is as follows.

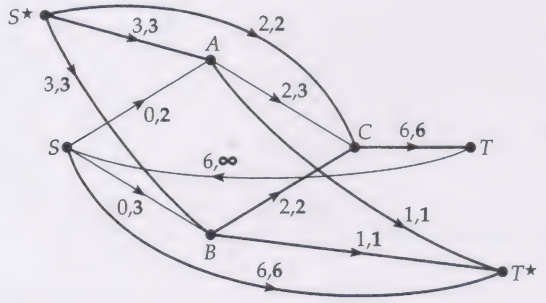


This flow saturates all the arcs out of S^* and all the arcs into T^*, so, by the feasibility theorem, the original network is feasible.

(c) The corresponding basic network N^* is as follows.

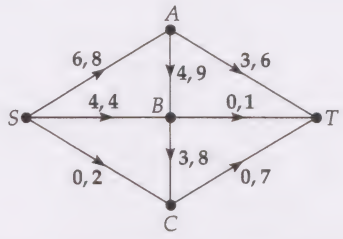


A suitable flow from S^* to T^* is as follows.

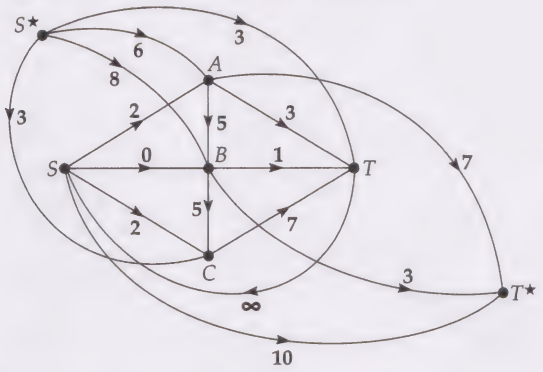


This flow saturates all the arcs out of S^* and all the arcs into T^*, so, by the feasibility theorem, the original network is feasible.

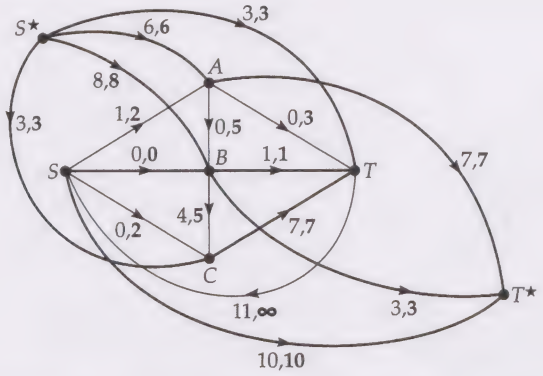
4.5 The original network is as follows.



The corresponding basic network N^* is as follows.

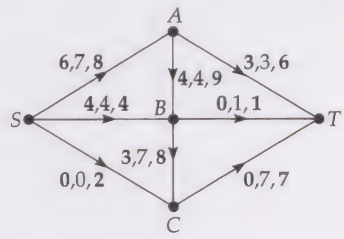


A suitable flow from S^* to T^* is as follows.



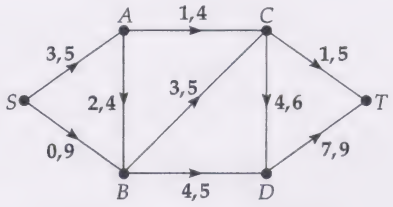
This flow saturates all the arcs out of S^* and all the arcs into T^*, so, by the feasibility theorem, the original network is feasible.

The corresponding feasible flow in the original network is:

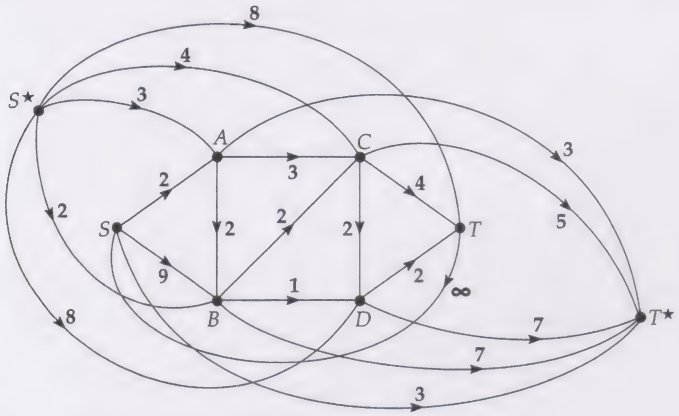


The flow along the arc SAT can now be increased by 1 to give the maximum flow found in the solution to Exercise 4.2.

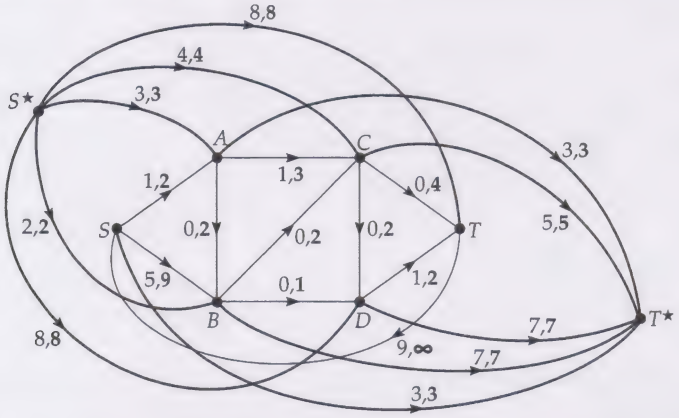
4.6 The original network is as follows.



The corresponding basic network N^* is as follows.

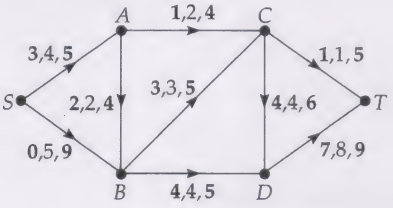


A suitable flow from S^* to T^* is as follows.



This flow saturates all the arcs out of S^* and all the arcs into T^*, so, by the feasibility theorem, the original network is feasible.

The corresponding feasible flow in the original network is:



The flow can be increased as follows to give the maximum flow found in the solution to Exercise 4.3: along *SACT* by 1, along *SBCT* by 2, and along *SBDT* by 1.

Solutions to the problems

Solution 1.1

(a), (c) and (f) are cutsets;

(b) is not a cutset, since removal of its edges does not disconnect the graph;

(d) is not a cutset, since we can disconnect the graph by removing yt;

(e) is not a cutset, since we can disconnect the graph by removing xz and yz.

Solution 1.2

(a) $\lambda(G) = 2$ (for example, remove the edges vw and xy);

(b) $\lambda(G) = 1$ (remove any edge);

(c) $\lambda(G) = 3$ (for example, remove the edges uw, ux and vx).

Note that if G is *any tree* with at least two vertices then $\lambda(G) = 1$.

Solution 1.3

(a) and (d) are vertex cutsets;

(b) is not a vertex cutset, since removal of its edges does not disconnect the graph;

(c) is not a vertex cutset, since we can disconnect the graph by removing u and x, or by removing y.

Solution 1.4

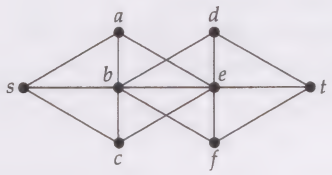
(a) $\kappa(G) = 2$ (for example, remove the vertices v and x), $\lambda(G) = 2$, $\delta(G) = 2$;

(b) $\kappa(G) = 1$ (for example, remove the vertex v), $\lambda(G) = 1$, $\delta(G) = 1$;

(c) $\kappa(G) = 2$ (for example, remove the vertices w and x), $\lambda(G) = 3$, $\delta(G) = 3$.

Note that if G is *any tree* with at least two vertices then $\kappa(G) = 1$. A tree with two vertices is K_2.

Solution 1.5



In each case there are several possibilities, for example:

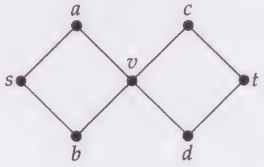
(a) *saet*, *sbdt*, *sceft*; (b) *sbet*, *sabdt*; (c) *saet*, *sbft*.

This graph does not contain three vertex-disjoint st-paths, since every st-path must pass through at least one of the two vertices b and e.

Solution 1.6

(a) If two vertex-disjoint st-paths were not edge-disjoint, then they would have an edge in common. But this would mean that they had at least one vertex (other than s and t) in common, contradicting the fact that they are vertex-disjoint.

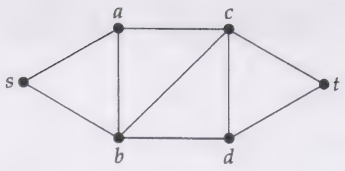
(b) There are many possibilities, for example:



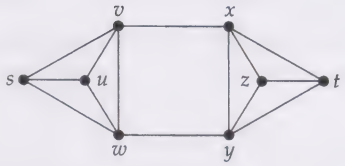
In the above graph, the only pairs of edge-disjoint st-paths are *savct* and *sbvdt*, and *savdt* and *sbvct*. In neither case are the paths vertex-disjoint, since they all pass through the vertex v.

Solution 1.7

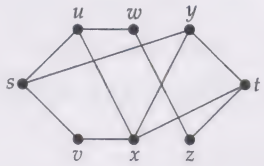
(a) In this case, $k = 2$; two edge-disjoint st-paths are $sact$ and $sbdt$, and two edges separating s from t are sa and sb. Thus the maximum number of edge-disjoint st-paths and the minimum number of edges separating s from t are both equal to 2.



(b) Again, $k = 2$; two edge-disjoint st-paths are $svxt$ and $swyt$, and two edges separating s from t are vx and wy. Thus the maximum number of edge-disjoint st-paths and the minimum number of edges separating s from t are both equal to 2.

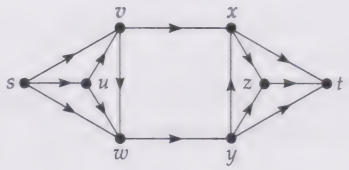


(c) In this case, $k = 3$; three edge-disjoint st-paths are $suwzt$, syt and $svxt$, and three edges separating s from t are su, sv and sy. Thus the maximum number of edge-disjoint st-paths and the minimum number of edges separating s from t are both equal to 3.

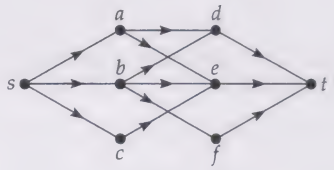


Solution 1.8

(a) In this case, $k = 2$; two arc-disjoint st-paths are $svxt$ and $swyt$, and two arcs separating s from t are vx and wy. Thus the maximum number of arc-disjoint st-paths and the minimum number of arcs separating s from t are both equal to 2.

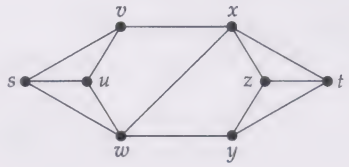


(b) In this case, $k = 3$; three arc-disjoint st-paths are $sadt$, $sbft$ and $scet$, and three arcs separating s from t are sa, sb and sc. Thus the maximum number of arc-disjoint st-paths and the minimum number of arcs separating s from t are both equal to 3.



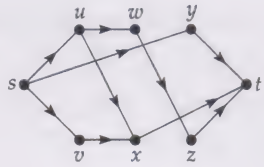
Solution 1.9

In this case, $k = 2$; two vertex-disjoint st-paths are $svxt$ and $swyt$, and two vertices separating s from t are v and w. Thus the maximum number of vertex-disjoint st-paths and the minimum number of vertices separating s from t are both equal to 2.



Solution 1.10

In this case, $k = 3$; three vertex-disjoint st-paths are $suwzt$, syt and $svxt$, and three vertices separating s from t are u, v and y. Thus the maximum number of vertex-disjoint st-paths and the minimum number of vertices separating s from t are both equal to 3.



Solution 1.11

(a) For C_n, we have

$$\kappa(C_n) = 2 \quad \text{and} \quad 2m/n = 2n/n = 2,$$

so C_n has optimal connectivity.

(b) For K_n, the number of edges is $\frac{1}{2}n(n-1)$, by a consequence of the handshaking lemma. We have

$$\kappa(K_n) = n - 1$$

and

$$2m/n = 2 \times \tfrac{1}{2}n(n-1)/n = n - 1,$$

so K_n has optimal connectivity.

See *Graphs 1*, Theorem 1.2.

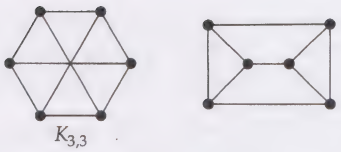
(c) For $K_{r,r}$, we have

$$\kappa(K_{r,r}) = r \quad \text{and} \quad 2m/n = 2r^2/(r+r) = r,$$

so $K_{r,r}$ has optimal connectivity.

Solution 1.12

(a) There are only two regular graphs with 6 vertices and 9 edges:



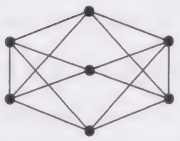
$K_{3,3}$

For each of these two graphs, we have

$$\kappa(G) = 3 \quad \text{and} \quad 2m/n = 3,$$

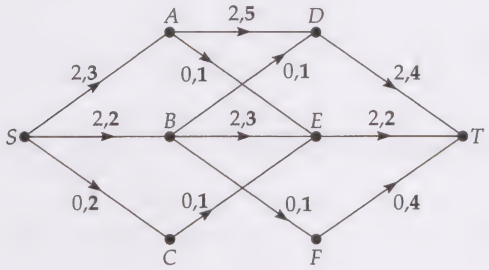
so the graphs have optimal connectivity.

(b) There is only one possibility — shown in the margin.



The removal of the middle three vertices disconnects the graph, so $\kappa(G) = 3$; since $\delta(G) = 4$, the graph does not have optimal connectivity.

Solution 2.1



(a) First we check that the feasibility condition is satisfied:

arc SA:	$2 \le 3$;	arc AE:	$0 \le 1$;	arc CE:	$0 \le 1$;
arc SB:	$2 \le 2$;	arc BD:	$0 \le 1$;	arc DT:	$2 \le 4$;
arc SC:	$0 \le 2$;	arc BE:	$2 \le 3$;	arc ET:	$2 \le 2$;
arc AD:	$2 \le 5$;	arc BF:	$0 \le 1$;	arc FT:	$0 \le 4$.

Next we check that the flow conservation law is satisfied:

vertex A: total flow in = 2 = total flow out;
vertex B: total flow in = 2 = total flow out;
vertex C: total flow in = 0 = total flow out;
vertex D: total flow in = 2 = total flow out;
vertex E: total flow in = 2 = total flow out;
vertex F: total flow in = 0 = total flow out.

(b) The total flow out of S is $2 + 2 + 0 = 4$;
the total flow into T is $2 + 2 + 0 = 4$.

Since none of the flow is lost along any arc or at any vertex, everything which leaves S must end up at T, and so the total flow out of S is equal to the total flow into T.

Note that this result can be expressed in the form:

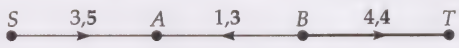
> the sum of the 'out-flows' of the vertices is equal to the sum of the 'in-flows'.

This is because, for each vertex other than S and T, the out-flow must equal the in-flow, and so *the out-flow at S is equal to the in-flow at T.*

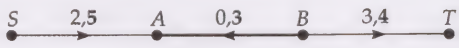
This result can be regarded as a network analogue of the handshaking dilemma. (See *Graphs 1*, Section 3.2.)

Solution 2.2

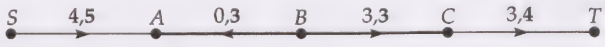
(a) The flow can be increased by 2, thus:



(b) The flow can be increased by 1, thus:



(c) The flow can be increased by 3, thus:



Solution 2.3

(a) First, for each of the *forward* arcs, calculate the largest amount by which the flow can be *increased* without exceeding the capacity, and let the smallest of these numbers be r.

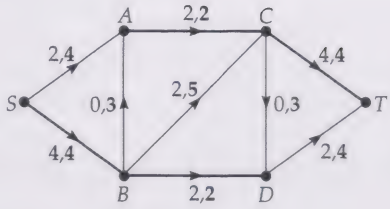
Next, for each of the *backward* arcs, calculate the largest amount by which the flow can be *decreased* without becoming negative, and let the smallest of these numbers be s.

The required amount is the smaller of these numbers r and s.

(b) For the given flow-augmenting path,
r is 3, the smallest of the numbers 3, 3 and 7;
s is 2, the smallest of the numbers 3, 2 and 5.
The required amount is therefore 2.

Solution 2.4

There are several ways of achieving a maximum flow of value 6. One such is shown in the margin.

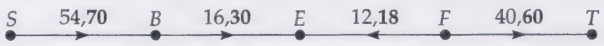


This can be obtained by starting with the zero flow and:

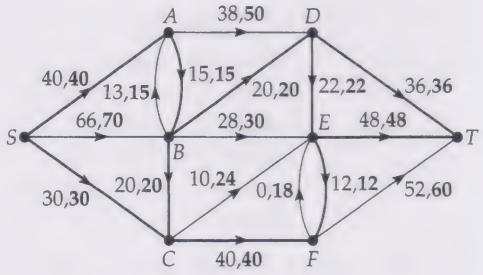
increasing the flow along $SACT$ by 2;
increasing the flow along $SBDT$ by 2;
increasing the flow along $SBCT$ by 2.

Solution 2.5

No. A flow-augmenting path is $SBEFT$:



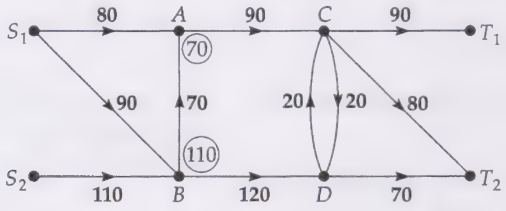
The flow along this path can be increased by 12 to give the flow of value 136 shown in the margin.



There are no further flow-augmenting paths in this network, so the above flow of value 136 is a maximum flow.

Solution 2.6

We replace the undirected edge CD of capacity 20 by two arcs CD and DC, each of capacity 20:

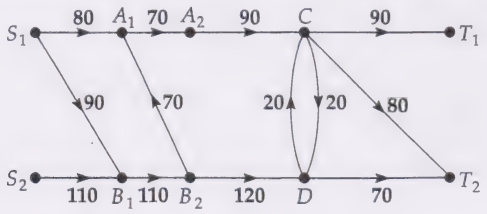


Solution 2.7

We replace the vertex A of capacity 70 by an arc A_1A_2 of capacity 70.

We replace the vertex B of capacity 110 by an arc B_1B_2 of capacity 110.

We then adjust the arcs into and out of A and B, as shown below.

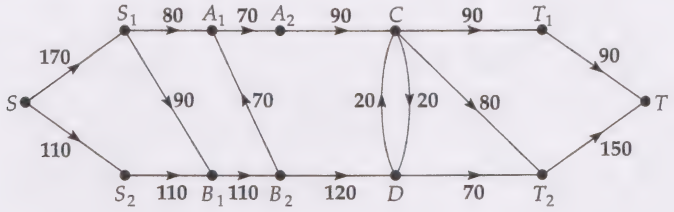


Solution 2.8

We add a super-source S, a super-sink T and the new arcs:

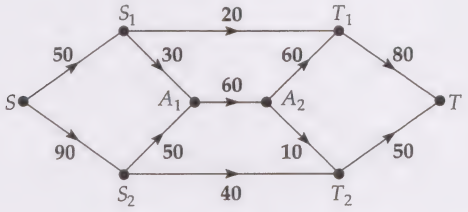
SS_1 with capacity 80 + 90 = 170;
SS_2 with capacity 110;
T_1T with capacity 90;
T_2T with capacity 80 + 70 = 150.

We thus obtain the following basic network.

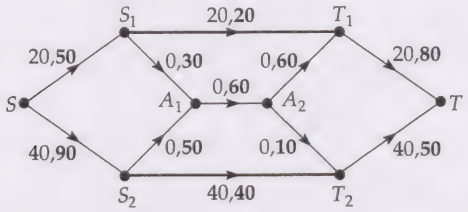


Solution 2.9

STEP 1 Applying the second and third transformations described in the text, we obtain the following basic network.

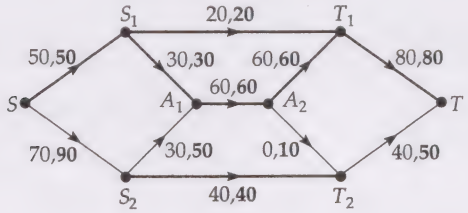


STEP 2 We can send a flow of value 20 along the flow-augmenting path SS_1T_1T and a flow of value 40 along the flow-augmenting path SS_2T_2T, as follows.



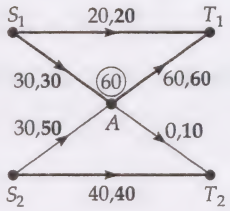
Next, we can send a flow of value 30 along the flow-augmenting path $SS_1A_1A_2T_1T$ and a flow of value 30 along the flow-augmenting path $SS_2A_1A_2T_1T$.

We thus obtain the following flow.



The arcs S_1T_1, A_1A_2 and S_2T_2 are all saturated, so the value of the flow cannot be increased any further; thus the value of a maximum flow is 120.

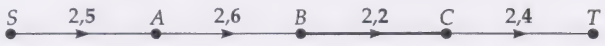
STEP 3 This gives rise to the maximum flow of value 120 in the original network, shown in the margin.



Note that other maximum flows (that is, flows of value 120) are possible, for example the flows of 60 and 0 through the arcs AT_1 and AT_2 can be replaced by flows of 50 and 10, respectively.

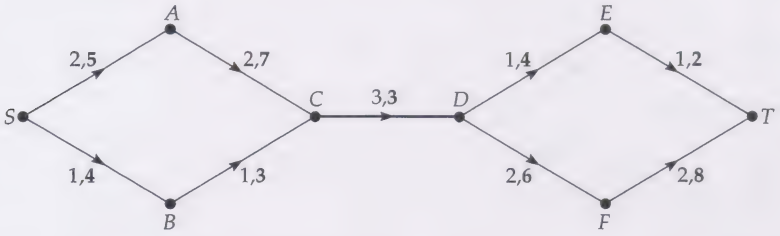
Solution 3.1

(a) The value of the maximum flow is 2:



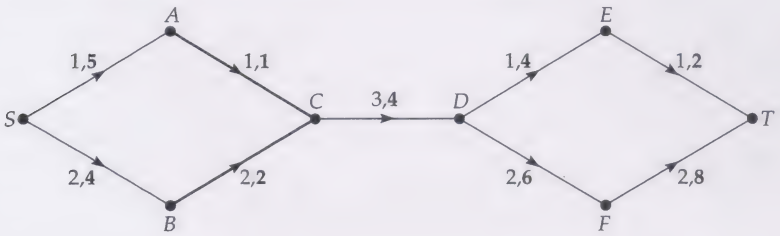
The worst bottleneck is the arc BC, which restricts the value of every flow to at most 2.

(b) The value of a maximum flow is 3; for example:



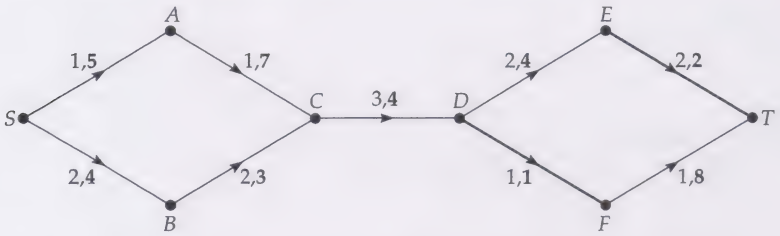
The worst bottleneck is the arc CD, which restricts the value of every flow to at most 3.

(c) The value of a maximum flow is 3; for example:



The worst bottleneck comprises the two arcs AC and BC, which restricts the value of every flow to at most $1 + 2 = 3$.

(d) The value of a maximum flow is 3; for example:



The worst bottleneck comprises the two arcs DF and ET, which restricts the value of every flow to at most $1 + 2 = 3$.

Solution 3.2

(a) The minimum cut consists of the arc BC, with capacity 2;

$X = \{S, A, B\}, \quad Y = \{C, T\}$.

(b) The minimum cut consists of the arc CD, with capacity 3;

$X = \{S, A, B, C\}, \quad Y = \{D, E, F, T\}$.

(c) The minimum cut comprises the arcs AC and BC, with capacity 3;

$X = \{S, A, B\}, \quad Y = \{C, D, E, F, T\}$.

(d) The minimum cut comprises the arcs DF and ET, with capacity 3;

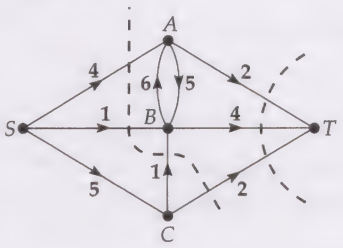
$X = \{S, A, B, C, D, E\}, \quad Y = \{F, T\}$.

Solution 3.3

	arcs in cut	X	Y	capacity of cut
1	SA, SB, SC	S	A, B, C, T	$4+1+5=10$
2	SC, SB, BA, AB, AT	S, A	B, C, T	$5+1+5+2=13$
3	SA, BA, AB, BT, CB, SC	S, B	A, C, T	$4+6+4+5=19$
4	SA, SB, CB, CT	S, C	A, B, T	$4+1+1+2=8$
5	SC, CB, BT, AT	S, A, B	C, T	$5+4+2=11$
6	AT, AB, BA, SB, CB, CT	S, A, C	B, T	$2+5+1+1+2=11$
7	SA, BA, AB, BT, CT	S, B, C	A, T	$4+6+4+2=16$
8	AT, BT, CT	S, A, B, C	T	$2+4+2=8$

Remember that arcs directed from a vertex in Y to a vertex in X are not included when calculating the capacity of a cut.

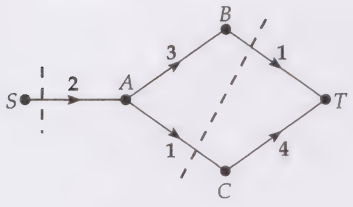
The minimum cuts are those with capacity 8: cuts (4) and (8). They are drawn on the following diagram.



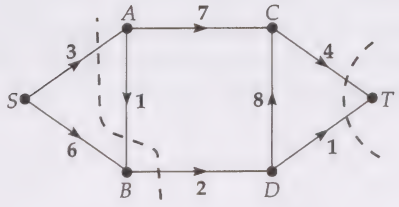
Solution 3.4

There are many possible answers, the simplest of which is:

Other examples are:

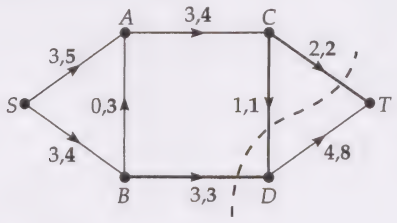


and

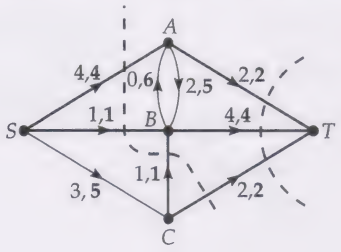


Solution 3.5

(a) We already know from Example 3.2 that the arcs BD, CD and CT form a minimum cut of capacity 6, so it follows that the value of any flow cannot exceed 6. The following diagram shows a flow of value 6, which must therefore be a maximum flow.



(b) We already know from the solution to Problem 3.3 that {*SA*, *SB*, *CB*, *CT*} and {*AT*, *BT*, *CT*} are minimum cuts of capacity 8, so it follows that the value of any flow cannot exceed 8. The following diagram shows a flow of value 8, which must therefore be a maximum flow.



Solution 3.6

(a) Its value must be 7 or less.

(b) Its capacity must be 11 or more.

(c) The value of a maximum flow and the capacity of a minimum cut must lie between 4 and 6 (inclusive).

(d) You have made a mistake in your calculations (because the value of a flow cannot exceed the capacity of a cut).

Solution 3.7

The arcs *CF*, *FE*, *EF*, *ET* and *DT* form a cut of capacity 136. Since we have found a flow of value 136 and a cut of capacity 136, by the rule preceding these problems, the flow must be a maximum flow.

Solution 3.8

(a) The statement is FALSE. To see this, consider the example of Problem 3.7 in which the minimum cut {*CF*, *FE*, *EF*, *ET*, *DT*} contains the unsaturated arc *FE*. A correct statement is:

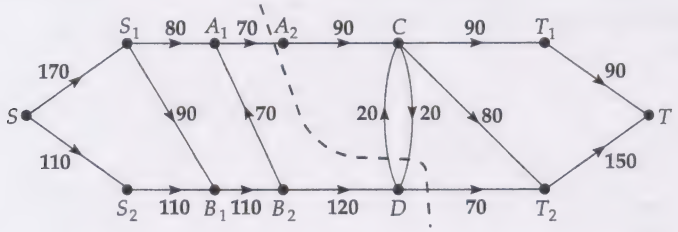
> every minimum cut dividing a network carrying a maximum flow into two parts *X* and *Y* consists entirely of saturated arcs directed from *X* to *Y* and arcs directed from *Y* to *X* carrying a zero flow.

(b) The statement is TRUE. Since the capacity of every arc is an integer, the capacity of every cut is an integer (since it is a sum of integers). In particular, the capacity of any minimum cut is an integer, and so, by the max-flow min-cut theorem, the value of any maximum flow is also an integer.

Solution 3.9

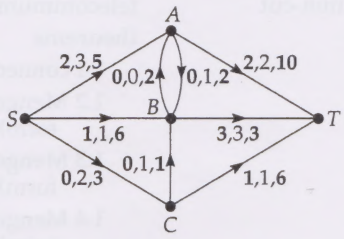
We know from Section 2.3 that a maximum flow in this network has value 160, so we are looking for a minimum cut of capacity 160.

This maximum flow saturates the arcs A_1A_2, DC and DT_2, so a minimum cut is $\{A_1A_2, DC, CD, DT_2\}$ with capacity 70 + 20 + 70 = 160.



Solution 4.1

First we check that the feasibility condition is satisfied:



arc SA: $2 \le 3 \le 5$ arc AB: $0 \le 1 \le 2$ arc AT: $2 \le 2 \le 10$;
arc SB: $1 \le 1 \le 6$ arc BA: $0 \le 0 \le 2$ arc BT: $3 \le 3 \le 3$;
arc SC: $0 \le 2 \le 3$ arc CB: $0 \le 1 \le 1$ arc CT: $1 \le 1 \le 6$.

Next we check that the flow conservation law is satisfied:

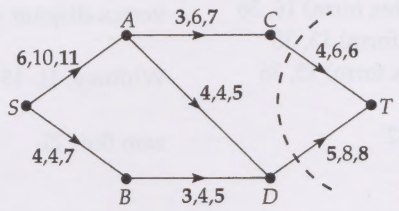
vertex A: total flow in = 3 = total flow out;
vertex B: total flow in = 3 = total flow out;
vertex C: total flow in = 2 = total flow out.

The value of the flow is equal to the flow out of S (or the flow into T), which is 6.

Solution 4.2

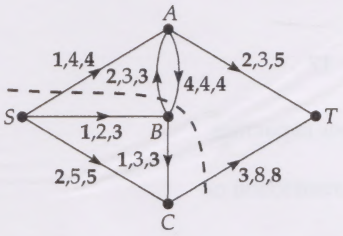
(a) The arcs CT and DT form a minimum cut of capacity $6 + 8 = 14$.

A maximum flow of value 14 is shown in the following diagram.

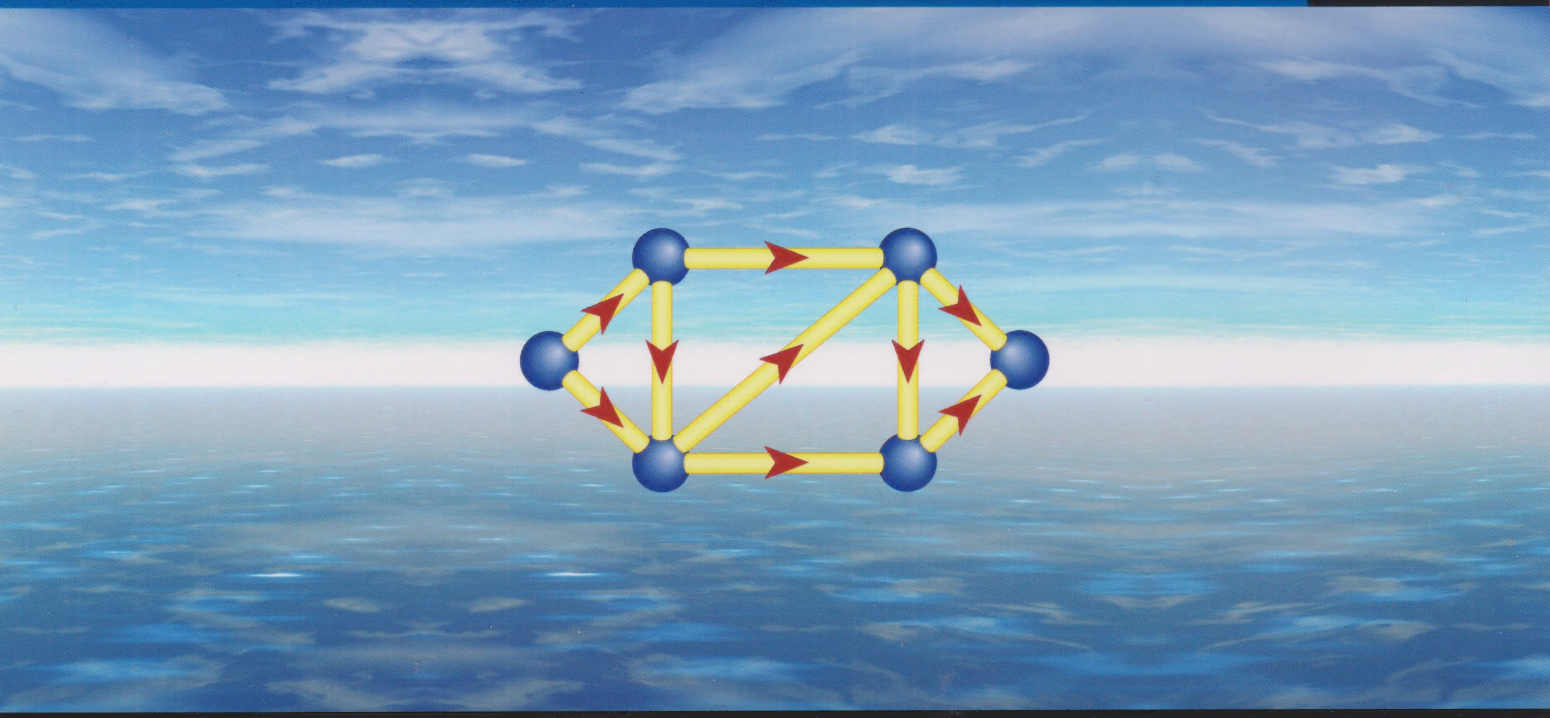


(b) The arcs SA, BA, AB and CT form a minimum cut of capacity $(4 + 3 + 8) - 4 = 11$.

A maximum flow of value 11 is shown in the following diagram.



Index



MT365 Graphs, networks and design

The Open University

MT365 Networks 1
ISBN 978 0 7492 5422 3

9 780749 254223